Using Cryptographic and Watermarking Algorithms

Jana Dittmann
Fraunhofer Institute Integrated Publication and Information Systems, Germany

Petra Wohlmacher
University of Klagenfurt, Austria

Klara Nahrstedt
University of Illinois, Urbana–Champaign

Because of multimedia's structure and complexity, security mechanisms for multimedia data should be specific for each purpose. We introduce the most important security requirements for all types of multimedia systems. We also survey revocation methods for digital certificates and introduce a media-independent classification scheme.

Recently, security has become one of the most significant and challenging problems for spreading new information technology. Because we can easily copy digital data, multiply it without information loss, and manipulate it without detection, security solutions are increasingly important. Whether or not a multimedia system is sufficiently secure will have a substantial influence on its acceptance. Security solutions that address fields such as distributed production and e-commerce are especially necessary because they provide access control mechanisms to prevent misuse and theft.

To assess the trustworthiness of information technology systems, researchers have published catalogs for security criteria.[1-3] One of the most important is the European Information Technology Security Evaluation Criteria (ITSEC) catalog of criteria,[2] which evaluates IT system security. This catalog defines security criteria within different classifications regarding three basic threats:

- confidentiality (unauthorized information revealing),
- integrity (unauthorized data modification), and
- availability (unauthorized withholding of information or resources).

These threads, applied to multimedia systems, require careful analysis of security requirements, clear understanding of multimedia-related security problems, and scalable and flexible solutions to protect multimedia systems against incoming attacks. In this article, we present the most important security requirements and measures, derived from these threads and discuss problems with securing multimedia data. We also describe various security solutions such as security mechanisms and revocation methods for digital signatures, a media-independent watermarking classification scheme for copyright protecting multimedia data, and a secure end-to-end resource reservation protocol for protecting Quality of Service parameters needed during multimedia system setup.

Requirements and measures

Security requirements are met by security measures, which generally consist of several security mechanisms that can implement security services. Overall, a security policy describes security requirements. The security policy also defines which measures realize these requirements.

The following security requirements, which use cryptographic mechanisms and digital watermarking techniques, are essential for multimedia systems:

- *Confidentiality*. Cipher systems keep information secret from unauthorized entities.
- *Data integrity*. One-way hash functions, message authentication codes, digital signatures (especially content-based digital signatures), fragile digital watermarking, and robust digital watermarking can detect data alteration.
- *Data-origin authenticity*. Message authentication codes, digital signatures, fragile digital watermarks, and robust digital watermarks enable proof of origin.
- *Entity authenticity*. Authentication protocols ensure that an entity is the one it claims to be.

1070-986X/01/$10.00 © 2001 IEEE

- *Nonrepudiation*. Nonrepudiation mechanisms prove to involved parties and third parties whether or not a particular event occurred or a particular action happened. The event or action can be generating, sending, receiving, and submitting or transporting a message. Nonrepudiation certificates, nonrepudiation tokens, and protocols establish information accountability. These mechanisms are based on message-authentication codes or digital signatures combined with notary services, time-stamping services, and evidence recording.

Security mechanisms

Security mechanisms for multimedia systems build on cryptographic mechanisms and digital watermarking techniques. Researchers have proposed a variety of watermarking techniques, but it's difficult to classify the approaches and measure their quality. Here we concentrate on the problems with multimedia data derived from applying cryptographic mechanisms.

Cryptographic mechanisms

Modern cryptographic mechanisms are mainly based on different, unproven assumptions, concerning easy and hard computation of functions. Intuitively, we can describe easy computation as a problem that we can solve in an acceptable time period and hard computation as one we can't solve in an acceptable time period using all of today's available resources. In cryptography, we use two types of these functions: one-way functions and trapdoor one-way functions.[4]

We implement the most important cryptographic mechanisms with cryptosystems, which consist of two sets of functions: a set of keys (they parameterize these functions) and sets (on which these functions operate). Cryptosystems are subdivided into private-key and public-key cryptosystems. In private-key cryptosystems, the communicating entities share a key K, called the secret key, which must be kept secret. The size of the key space must be large enough to make it hard to find the right key K.

Public-key cryptosystems are based on trapdoor one-way functions. Each entity holds a key pair (PK, SK). This pair consists of a private key SK and a public key PK corresponding to SK. The key SK must remain secret, but the key PK can be public. Given a public key PK, it's computationally infeasible to find the private key SK if the trapdoor information is unknown. In other words, even with the most powerful computers, it's computationally infeasible to deduce PK from SK.

Usually, cryptographic mechanisms take each bit of data for input to calculate the output, which we need to provide a security mechanism. Additionally, if one bit of the input or output changes, the encryption or even the validation will fail in most cases.

Multimedia applications require high performance, and their data can be altered due to transmission errors, high compression rates, or scaling operations. Therefore, it's difficult to define a suitable input for cryptographic mechanisms. If we apply cryptographic mechanisms directly to all media data, some problems may occur.

Digital watermarking

Digital watermarking techniques based on steganographic systems can embed information directly into the media data. Watermarking represents an efficient technology for ensuring data integrity and data-origin authenticity. Watermarking techniques, used for digital imagery, audio, and 3D models, are relatively young but are growing at an exponential rate. We can embed copyright, customer, or integrity information into the media data as transparent patterns using a secret key. Because we integrate the security information into the media data, we can ensure the confidentiality of security information using a secret key but not the confidentiality of the media data itself.

Security measures

Security measures are integral and important parts of any security solutions because they implement and enforce security requirements, defined by security policies. In accordance with the security requirements we specified, such as confidentiality, data integrity, data-origin authenticity, entity authenticity, and nonrepudiation, we present their corresponding security measures and trade-offs when applied to multimedia systems.

Confidentiality

We can achieve confidentiality by using cipher systems, which keep information secret from unauthorized entities. We can use private-key and some public-key cryptosystems for cipher systems. Because of performance considerations, large amounts of data are enciphered by a session-key scheme—also known as a hybrid cryptosystem because it applies both private-key and public-key cryptosystems. None of these mechanisms, however, provides protection after deciphering, such as checking if the data has changed or the identity of

the data's owner to ensure copyright protections.

Within streaming applications, we must usually transmit a huge amount of data from a sender to a receiver in a time-critical and confident manner. Even the session-key scheme fails to support the necessary performance level. Another problem derived from such large amounts of data is that transmission errors, high compression rates, or scaling operations during transmission can alter media data. Using common encryption methods, decrypting a cipher-text block might fail. With these methods, we can't recover the original plain text if one bit of the cipher-text block is altered.

A general solution to these problems is partial encryption. Instead of encrypting all the data, we only encipher special parts of the data. If we select well, we can achieve a sound confidentiality of all the data.

Considering the results from performance measures in secure video systems, researchers have proposed several methods for partial encryption of video data in the last few years.[5-7] The basic idea behind these approaches is to encrypt only relevant information—for example, motion vectors, coefficients, or header information. MPEG-1, MPEG-2, H.261, and H.263 are widespread compression standards used in most videoconferencing applications. They're well suited for partial encryption because they use the discrete cosine transform (DCT), which has a high potential for partitioning data into more or less relevant parts (entropy of the coefficients). Also, they encode large amounts of video data by referencing preceding or succeeding blocks (intracoded blocks), protecting only the referenced blocks.

Several sophisticated approaches exist for applying partial encryption to nonscalable, standard-based hybrid video coding schemes such as MPEG video. Base-layer encryption doesn't require content parsing and, therefore, has a much lower overall computational complexity than partial MPEG encryption. For base-layer encryption, we must determine the amount of encrypted data a priori whereas partial MPEG encryption allows different security levels even if a video is already encoded.

Data integrity

We can check the data's integrity with one-way hash functions. Furthermore, we can apply some mechanisms discussed in the next section to detect data alteration. Detection mechanisms can't prevent data manipulations, but they make these manipulations detectable. The protected data remain in plain text. A hash function maps strings of arbitrary length to strings of a maximum or fixed length. Hash functions are public—that is, they don't use secret information to compute a hash value. Thus, everyone knows the function to compute the hash value and can check the data's integrity.

In multimedia applications, we can change media data with compression or scaling without content manipulation. Therefore, hash functions aren't appropriate if they're applied to media data directly. To solve the problem, we should apply hash functions to data concerning the media stream's semantics. These data are feature codes, which represent the media's content. For image data, we can use DCT coefficients or edges.[8,9]

Data-origin authenticity

Message authentication codes (MACs), digital signatures (especially content-based digital signatures), fragile digital watermarks, and robust digital watermarks ensure data-origin authenticity. Additionally, the first three mechanisms also ensure data integrity. Similar to data-integrity mechanisms, all four are detection mechanisms and the protected data remains in plain text.

A MAC is a one-way hash function that's parameterized by a secret key. Only those entities that know the secret key can calculate the MAC.

We can use public-key cryptosystems to generate and verify digital signatures.[10,11] A digital signature of an entity A (the signer) of data m depends on m and the private key of A. Each user can verify the authenticity of the signature created by A within a verification process using the public key of A.

Regarding multimedia data, we can use digital signatures for image and video authentication to ensure trustworthiness with public-key cryptosystems. However, applying digital signatures directly to digital image data is vulnerable to image-processing techniques like conversion, compression, or scaling, which can irreversibly change the image material without content modifications. Although the image's content hasn't changed and the viewers still have the same image impression, the signature verification will fail. Manipulations can be content-preserving or content-changing. We should apply digital signatures to the media data's feature codes. We must use feature codes that aren't altered by allowed operations, such as scaling and conversion of media formats. Because feature codes should represent the media's content, we call these mechanisms content-based authentication codes or content-based digital signatures.

Media integrity using digital watermarks differs from the introduced cryptographic mechanisms of hash functions, message authentication codes, digital signatures, and content-based digital signatures, where the check value is appended to the data. Watermarking uses redundant information in media data to slightly modify the media and embed integrity information. The integrity verification data is embedded in the media rather than appended to it. Possessing the appropriate secret key *K*, we can verify the watermark and evaluate whether the data was altered (particularly tampered) by checking the embedded information.

For the moment, using public-key cryptosystems for only watermarking is unknown. Several techniques and concepts have been introduced[12] for image and audio data such as fragile digital watermarks using private-key cryptosystems. The existing approaches have different strategies for tamper detection. Some approaches sense changes such as modifications in check values, and others try to recognize only content changes.[8,9] The latter approaches are usually called content-fragile watermarks. The problem with embedding the content as a watermark is that watermarking techniques usually can't embed more than 10 to 100 bytes. Therefore, it's impossible to embed the content with a data rate higher than 1 Kbyte. The solution for content-fragile watermarks combines a robust watermarking technique with the content characteristic for integrity detection. The main idea is to initialize a robust watermarking pattern with the media's content.

A robust mark is designed to resist attacks that attempt to remove or destroy it. The intention is to embed owner, producer, or customer identification into the media data to ensure copyrights using a private-key cryptosystem. Similar to fragile watermarks, public-key techniques also aren't applied for robust watermarks. The robust watermark should remain present even after media processing or attacks, even if the content is manipulated.

Entity authenticity

Often it's necessary to ensure the authenticity of entities—that is, to guarantee that the communicating parties (people or devices) are who or what they claim to be. Schemes that enable such proof are authentication protocols. The simplest version of an authentication protocol is the challenge-response protocol that works as follows: The verifier sends to the claimant a randomly generated number, or a challenge. The claimant returns a response to the verifier that consists of a value generated by using the challenge and a secret key. The verifier can prove that the claimant possesses the secret key. For each authentication, the verifier generates a new question. We can implement these protocols based on private- or public-key cryptosystems.[13]

Within legal facilities, digital signatures on their own are insufficient to link data and actions to their originators.

Nonrepudiation

Within legal facilities, digital signatures on their own are insufficient to link data and actions to their originators. These operations can use security infrastructures and techniques to provide some evidence that the courts will accept. Nonrepudiation mechanisms,[14] which are based on private-key cryptosystems (message authentication code) or public-key cryptosystems (digital signatures), support such security techniques. They consist of nonrepudiation certificates, nonrepudiation tokens, and protocols. Trusted third parties supply notary services, time-stamping services, and evidence recording. By means of these mechanisms, we can prove to involved parties and third parties whether or not a particular event occurred. These mechanisms are subdivided into nonrepudiation of origin, delivery, submission, and transport.

Digital certificates

Using public-key cryptosystems raises problems. For example, with session-key schemes, we can only recover the encrypted session key (and thus the plain text) with the recipient's private key (so-called addressed confidentiality). However, these schemes can't ascertain whether the public key, which is used to encrypt the session key, actually belongs to a particular person (or device).

Using digital signatures and signature-based authentication protocols, we can check whether the signature to particular data was generated by a specific key by verifying the digital signature. Thus, we can prove a message or communication's authenticity. However, we can't prove

whether the used keys actually belong to a certain person or device.

Obviously, we need an authentic link between the public key and its owner. Public-key certificates provide such a link.[15] For issuing certificates, we need a trustworthy authority, or trust center. Trust centers authenticate the link between users and their public keys and can provide further services like nonrepudiation, revocation handling, time stamping, auditing, and directory service. Within a trust center, special components provide these services. Each trust center, and even its components, comply with a security policy. This policy regulates, for example, generating and distributing certificates and how to ensure the availability of the services.

Today, researchers are developing and establishing large security infrastructures to meet security requirements. For example, public-key infrastructures (PKI) together with public-key certificates and even attribute certificates form a basis that lets entities trust each other.

A special trustworthy authority named the certification authority generates and issues certifications. An authority's security policy describes the life-cycle of key pairs or attributes, respectively, and thus, a certificate's validity period. Within this period, we can be confident in its reliability. The validity of the public key or attribute is specified in the certificate and signed together with other data by the certification authority. Therefore, we can detect forged certificates. Usually certificates are submitted to an authority that provides certificates. The authority is mostly called a directory. A certificate's validity period is between several months and two years, but in some circumstances, a certificate must be revoked sooner than assigned.

The revocation management should be clearly defined for certification authorities, directories, and users. Certification authorities must provide a revocation service in a trustworthy manner and, therefore, publish a proper security policy. Users should know how and when a revocation must be initiated and how they get informed about a revocation. The certificate's owner or an authorized representative (which is already mentioned in the certificate or by a certification authority) initiates the revocation. Only the certification authority revokes certificates and complies with a revocation request when the initiators can prove their authorization. Usually, the certification authority submits the status of all certificates to a directory that answers users' requests concerning the certificates' validity. Depending on the specific security policy, another authority might also provide this service.

Additionally, revocation methods must fulfill other requirements. A revocation must be fast, efficient, timely, and appropriate for large infrastructures. Because of that, it's necessary, for example, to reduce the number of time-consuming calculations concerning verification processes of a digital signature and to apply other mechanisms or to minimize the amount of data transmitted. It's also desirable that a method can suspend a certificate temporarily and reuse it.

To prove a certificate's validity, a user has to perform different tests. One of the most critical tests is to determine whether a certificate has been revoked. Usually, this means a user sends a request to a directory. That request contains at least a serial number that represents a unique identifier for each certificate. The response includes the serial number, status, date, and reason for revocation.

Classification and reasons for revocation

We can classify methods for revocation several ways:

- *By their way of checking*. A method can perform a check either offline or online, or sometimes apply both methods. Within an offline scheme, a certification authority precomputes the validity information and then distributes it to the requester by a nontrusted directory. Within an online scheme, a trusted directory, which performs a proof of validity during each request and provides up-to-date information, provides the status information online.

- *By their kinds of lists*. Negative (black) lists contain revoked certificates and positive (white) lists contribute valid certificates. Some methods combine both mechanisms.

- *By their way of providing evidence*. A method gives direct evidence if a certificate is mentioned in a positive or negative list, respectively. Then, the certification authority is supposed to revoke or not revoke it, respectively. A method gives indirect evidence if it can't find a certificate on a list and, therefore, assumes the contrary.

- *By their way of distributing information either via a push or pull mechanism*.

Because of several threats, important reasons exist why a certificate would need to be revoked:[16]

- *Key compromise*. The private key of the subject (user) or of the issuer (certification authority) has been compromised or is suspected to be compromised—for example, broken or stolen.

- *Change of affiliation*. Some information in the certificate about the subject or any other information is no longer valid.

- *Superseded*. The certificate is superseded and no further reasons are available.

- *Ceased operation*. The certificate is no longer needed for its assigned purpose.

Some further arguments exist why a certificate needs to be revoked. (We sorted these items in decreasing order according to their urgency.)

- *Algorithm compromise*. The signature algorithm the certification authority uses has been broken in general or the algorithm of the certified public key is compromised. This might be caused by new advances in algorithm theory, number theory, or computer capabilities.

- *Revocation of superordinated certificate*. A certificate that's part of each certification path leading to the dedicated certification authority is revoked.

- *Loss or defect of security token, or loss of password*. Either the certificate's subject has lost its physical equipment or its equipment is damaged. Also, a password or a PIN that protects the token from unauthorized access is lost.

- *Change of key usage*. The certified key can no longer be used for its assigned purpose.

- *Change of security policy*. The certification authority no longer works under its defined policy—for example, it ceases to support a certificate service.

Usually, the status information about a certificate includes reasons for a revocation.

Specific revocation methods

Different kinds of revocation methods include certificate revocation lists (CRLs), a certificate revocation system (CRS), certificate revocation trees (CRTs), and online certificate status protocol (OCSP). All these methods require an authentic verification key from the certification authority.

The International Telecommunications Union-Telecommunications (formerly CCITT) introduced CRLs together with X.509 certificates in 1988. Since the second edition of the X.509 Recommendation in 1993, revocation lists are based on an improved version 2 by ITU-T and ISO/IEC 38. A CRL is a negative list, giving indirect evidence, provided offline. CRLs are periodically issued, usually monthly. A CRL contains a list of serial numbers of revoked certificates together with their date of revocation, the date of its generation, and the latest date of the next issue. More information, such as reasons for the revocation, can be added. Finally, the issuing certification authority digitally signs the CRL. Thus, users can check the certificate's freshness and authenticity. CRLs are periodically sent to a directory.

Users requesting the certificate's validity receive a full CRL. Then, they check the actuality and verify the CRL's signature. If this succeeds, they determine whether the queried certificate is included in the CRL. If users can't find the serial number, they can assume the certificate is still valid.

CRLs are straightforward, easy to understand, and thus widely used. Because the certificates' validity period is long and the number of users is immense, CRLs can grow extremely large. Therefore, a great amount of data must be transmitted. The fact that a CRL is only up-to-date at their point of issuing has led to *delta-CRLs*. A delta-CRL is issued between two CRL updates. It includes only changes since the last-issued CRL and so enhances efficiency. Delta-CRLs contain sequence numbers that let users verify the completeness of CRL information.

Silvio Micali introduced the CRS[17] in 1995. His idea uses online and offline signatures.[18] He improved his idea in 1996[19] when he redefined CRS by revocation status, and in 1998, he got a patent[20] for this work. A CRS mixes positive and negative lists and, thus, gives direct evidence. This method treats the validity status of each certificate separately. Here, a user sending a query concerning a single certificate's validity will get a response containing short information about this certificate. Depending on the up-to-date time schedule, the system can either operate online or offline. The certification authority signs a list L concerning all valid serial numbers of certificates. A hash function is applied to two certified values for n times. In each timeperiod, the verifier can check the validity of a specific certificate by using a new published hash value together with list L.

Paul Kocher introduced CRTs, which are based on hash trees,[21] in 1998. CRTs are negative lists, but they also support information about unrevoked certificates (mixed form). Regarding the sorted set of revoked certificates, all valid certificates can be assigned to specific validity intervals. Thus, CRTs give direct evidence.

The Internet Engineering Task Force developed the OCSP,[22] which specifies a protocol that determines a certificate's current validity status online. OCSP is designed for X.509 certificates, but it also works with other kinds of certificates. The protocol can be used instead of or even together with CRLs if users require more timely information about the status. Information about the way to obtain a certificate's status can be included within the extension fields of a X.509 certificate.

Researchers have developed other methods to revoke certificates. Besides these examples, we must analyze if and in which way a revocation method is appropriate in accordance to its purpose. An important aspect of a decision is cost. High costs derive from a great amount of transmitted data that's needed to provide a proper revocation and from measures that provide the availability of timely data. With offline systems, the time period between two updates is often long, so the validity can't be exact. However, this is sufficient for some applications. Online systems, suitable for purposes where more timely information is necessary, are obviously more expensive. Another aspect is whether a revocation method is applicable for a storage equipment like smart cards or other security tokens. Finally, the knowledge about different revocation methods isn't widespread. Efficient and practicable methods are still needed and are a topic of today's research. A main requirement for new developments and new ideas is that they can easily be integrated in the widely used X.509 certificates.

Media-independent classification scheme

Digital watermarking is a technology capable of solving important practical security problems. It's a multidisciplinary field that combines media and signal processing with cryptography, communication theory, coding theory, signal compression, and the theory of human perception. Interest in this field has recently increased because of the wide spectrum of applications it addresses. Although researchers have proposed a wide variety of techniques, it's difficult to classify the approaches and measure their quality.

Our intention is to classify the different watermarking schemes and to find quality measures. Our classification scheme takes the application areas into account and shows which parameters and attacks are essential. In comparison with Pitas,[23] we don't analyze the algorithms' details and don't refer to the domain where the watermark is embedded. Our goal is to give the users a system (available over the Web) to find the appropriate watermarking function along with their parameters.

Application-based classification

We've identified the following, general watermarking classes based on application areas for digital watermarking:

- *Copyright watermarks* mark the data with an owner or producer identification.
- *Fingerprint watermarks* mark the data with customer identifications to track and trace legal or illegal copies.
- *Copy control or broadcast watermarks* ensure copyrights with customer rights protocols (for example, for copy or receipt control).
- *Annotation watermarks* embed annotations or descriptions of the data's value or content.
- *Integrity watermarks* ensure the data's integrity and recognize manipulations.

In our classification scheme, we don't consider watermarking as an information hiding technique that has a secure cover communication.

Watermarking parameter

The most important properties of digital watermarking techniques are robustness, security, imperceptibility and transparency, complexity, capacity and possibility of verification, and invertibility:

- Robustness describes whether the watermark can be reliably detected after media operations. Robustness doesn't include attacks on the embedding scheme based on the knowledge of the embedding algorithm or on the availability of the detector function. It means resistance to blind, untargeted modifications or common media operations. For example, the Stirmark and Mosaik tool[24] attack watermarking algorithms' robustness with geometrical distortions. For manipulation recognition, the watermark must be fragile to detect altered media.

Table 1. Important parameters and attacks.

Watermark	Parameter	Attacks
Copyright watermark	High robustness.	Mosaik attack[24]
	High security.	Stirmark attack[19,24]
	Imperceptible.	Geometrical attacks[28,29]
	Blind methods are usually more practicable.	Histogram attacks[10]
	Capacity should fit the needs for a rightful owner identification.	Template attacks[27]
		Forgery attacks[31]
	Verification process is usually private, but public can also be desirable.	Rightful ownership attacks (invertability)[25]
Fingerprint watermark	See copyright watermark.	See copyright watermark.
	Nonblind techniques are useful.	Coalition attack[32,33]
Copy control and broadcast watermark	See fingerprint watermark.	See copyright watermark.
	Low complexity required.	
Annotation watermark	Robustness is less important in most cases.	No interest exists in attacking the watermark in most cases.
	Security isn't usually important.	
	Blind methods are preferable with low complexity.	
	High capacity.	
	Verification process is usually private, but public may be desirable.	
Integrity watermark	See copyright watermark.	Forgery attacks[31]
	Robustness needed until the data's semantics is destroyed (semifragile, content fragile).	Rightful ownership attacks (invertability)[25]
		Attacks on the fragility[34]

- Security describes whether the embedded watermarking information can't be removed beyond reliable detection by targeted attacks based on a full knowledge of the embedding algorithm and the detector, except the key, and the knowledge of at least one watermarked data. Security includes procedural attacks, such as the IBM attack,[25] or attacks based on a partial knowledge of the carrier modifications due to message embedding[26] or embedded templates.[27] The security aspect also includes false-positive detection rates.

- Transparency relates to human sensory factors. A transparent watermark causes no artifacts or quality loss.

- Complexity is the effort and time we need to embed and retrieve a watermark. This parameter is essential if we have real-time applications. Another aspect addresses whether or not the original data must be present in the retrieval process. We need to distinguish between nonblind and blind watermarking schemes.

- Capacity is how many information bits we can embed. It also addresses the possibility of embedding multiple watermarks in one document in parallel.

- The verification procedure is whether we have a private verification like private-key functions or a public-verification possibility like the public-key algorithms in cryptography.

- Invertibility is the possibility of producing the original data during the watermark retrieval.

The parameter optimizations are mutually competitive, so we can't do them at the same time. If we want to embed a large message, we can't simultaneously require large robustness. A reasonable compromise is always a necessity. On the other hand, if robustness to strong distortion is an issue, the message that can be reliably hidden must not be too long.

Important parameters

Each of the five classes of watermarks has its own quality parameters and standards. In Table 1, we point out the general watermarking parameters for each of the five watermark classes. We can use these as general quality metrics. Our goal is to classify new and existing algorithms into this scheme. We plan to offer a Web interface where researchers and industry can register their algorithms in a classification database to provide user transparency.

Also in Table 1, we show possible attacks, which depend on the application area.[28] For

Digital watermarking is a multidisciplinary field that combines media and signal processing with cryptography, communication theory, coding theory, signal compression, and the theory of human perception.

example, the fingerprint watermark has to deal with the coalition attack. If we watermark the original with different user identifications, we produce different copies. Customers could work together by comparing their different copies to find and destroy the fingerprint watermark.[32,33]

Another problem we recognize is robustness and recognizing manipulations for the integrity watermark. Robustness must adapt to content-changing and content-preserving manipulation, which the watermarking algorithms must address.

The integrity watermark is a fragile watermark that is readily altered or destroyed when the host image is modified through a linear or nonlinear transformation. The sensitivity of fragile watermarks to modification leads to their use in image authentication. Fridrich[12] classifies watermarks into fragile, semifragile, robust, and self-embedding as a means for detecting both malicious and inadvertent changes to digital media.

Currently, fragile watermarks are sensitive to change and can detect every possible change in pixel values. Therefore, they can be useful for parties that want to verify that an image hasn't been edited, damaged, or altered since it was marked. But in many applications, we have to cope with several allowed postproduction editing processes, which don't manipulate the content of the image or video data. The semifragile schemes try to address this problem. These techniques are moderately robust and the value identifying the presence of the watermark (a correlation in most cases) can serve as a measure of tampering. The problem with these schemes is that we can't recognize whether the media's content or message was affected or manipulated. Therefore, we find approaches that can distinguish malicious changes from innocent image-processing operations. We can term such techniques visual content authentication.[12]

We must distinguish between content-preserving and content-changing manipulations. Most existing techniques use threshold-based techniques to decide visual integrity. The main problem is to face the variety of allowed content-preserving operations. As the literature shows, most algorithms address the compression problem. However, often scaling, format conversion, or filtering are also allowed transformations, and most techniques recognize scaling, format conversion, or compression as integrity violation. To allow several postproduction editing processes, we need more sophisticated approaches.

No single scheme can have precise localization properties without being too sensitive. Depending on the application area, the user must choose the appropriate technique.

Securing RSVP for multimedia applications

Here, we give an example design for securing RSVP[35] for multimedia applications. Distributed multimedia applications require end-to-end QoS to be accepted and used. One approach is to provide end-to-end resource reservations. RSVP is a unicast and multicast signaling protocol for setting up network bandwidth reservation. RSVP sets up a distributed state in routers and hosts. A host uses it to request a specific QoS from the network for particular application data streams or flows. Routers also use it to deliver QoS requests to all nodes along the path(s) of the flows and to establish and maintain state to provide the requested service. The RSVP protocol raises several security issues:[35,36]

- *Message integrity and node* (router) *authentication*. Unauthorized people could corrupt or spoof up RSVP reservation request messages, leading to service theft or denial-of-service attacks. (To spoof up messages at network routers means to intercept messages by unauthorized users who then use the information to deceive and harm the message sender.)
- *User authentication*. Policy control will depend largely on the positive authentication of the user responsible for each reservation request.
- *Nonrepudiation*. Users shouldn't be able to falsely deny later that they sent messages.
- *Confidentiality*. The RSVP request messages mustn't be visible to outside parties. This implies a need for RSVP message encryption.

- *Replay attacks*. Intruders could replay the RSVP messages by causing wasteful reservations and/or service theft.

- *Traffic analysis*. Outside parties shouldn't be able to infer with RSVP requests by analyzing the traffic. This depends on the strength of the encryption mechanism used.

- *Cut-and-paste attacks*. Outside parties can copy the RSVP requests from one RSVP flow and paste it into another, leading to denial-of-service attacks or service theft.

Unauthorized people can achieve denial-of-service attacks by corrupting packets, degrading network use, spoofing up messages, and dropping packets.

The current solutions for securing RSVP suffer from high overhead[37] and scalability problems.[36] To make our design scalable, we divide the network into domains or subnetworks and modified the current algorithms.[36] In our design, each subnetwork has an ingress and egress node. All incoming traffic to a subnetwork enters through the ingress node, and all outgoing traffic goes through the egress node.

In this hierarchical network, we design a hybrid protocol, called the RSVP with scalable QoS protection (RSVP-SQoS) protocol, consisting of two major protocol-processing approaches: one within a subnetwork and the other across subnetworks. The security assumptions for the two approaches differ. Within a subnetwork, we assume a weaker security assumption than across subnetworks. So within a subnetwork, we use delayed integrity checking. When the RSVP messages go across subnetworks, we do a stronger integrity check with encryption, if necessary. The protocol authenticates with digital signatures to protect against replay attacks. The delayed integrity check within a subnetwork is done by making each egress node send a feedback message for integrity checking of RSVP QoS parameters within that subnetwork. While this integrity checking is going on, the egress node can wait (pessimistic approach) or forward the packet ahead (optimistic approach). Whenever it detects an intrusion, the system sends tear-down messages, tearing the connection down and preventing further intrusions.

Overall, RSVP-SQoS aims to provide a scalable and flexible solution, minimizing the delay in detecting intrusions with low overhead. We achieve the lower overhead in time and space with the differentiation of security measures within subnets and between subnets as opposed to hop-by-hop security measures.[37]

Conclusion

Despite numerous existing cryptographic and watermarking algorithms, applied to multimedia systems and presented in this article, many open questions and future research problems remain. We can categorize these questions and hence the future directions into at least three groups:

- encryption and multimedia processing and communication,

- copyright protection and multimedia coding, and

- secure protocols for multimedia delivery.

In the multimedia encryption area, we need to consider real-time algorithms for encryption because of the real-time nature of multimedia data during their processing and communication. This is a challenging problem because multimedia data can be large, segmented, and distributed; time for encryption and decryption can be limited, especially during the live video or audio playback; and the length and content of encryption keys may have to vary during the multimedia session to avoid various attacks.

In the area of copyright protection for multimedia data, we should consider new watermarking algorithms as new multimedia encoding and other image processing algorithms are being developed. New multimedia compression standards such as MPEG-4, MPEG-7, and others; new image and speech processing operations such as geometric transformations; and other multimedia operations present new challenges for developing robust and fragile watermarking schemes and preserving copyright protection.

In the security protocols area, we should consider problems such as real-time key management protocols and secure multicast protocols for multimedia distribution, secure protocols for multimedia-related parameters distribution, and collusion protection of security information in multicast applications such as pay-per-view or video-on-demand.

These problems and possible future research directions represent only a fraction of the many outstanding challenges that the multimedia community must solve. It's of great importance that we design and develop multimedia systems with secu-

rity and protection algorithms as an integral part of the overall system instead of as an afterthought. Otherwise, we expose multimedia systems to many highly undesirable security threads and performance overheads if we're aiming for high user acceptance of multimedia systems in our computing and communication environments. **MM**

Acknowledgments

We acknowledge and thank Vanish Talwar for his contribution in the area of RSVP-SQoS for multimedia applications.

We presented a preliminary version of parts of this article at the 2000 ACM Workshop on Multimedia and Security[38] and at the special session on Multimedia and Security at the IEEE International Conference on Multimedia and Expo (ICME) 2000.[39]

References

1. *Department of Defense Trusted Computer System Evaluation Criteria* (Orange Book), US Dept. of Defense, DOD 5200.28-STD, Dec. 1985.
2. "Information Technology Security Evaluation Criteria (ITSEC)," *Provisional Harmonised Criteria*, vol. 1.2, Jun. 1991.
3. *NATO Trusted Computer System Evaluation Criteria* (Blue Book), North Atlantic Treaty Organization, NATO AC/35-D/1027, 1987.
4. A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, Fla., 1997.
5. T. Kunkelmann, "Sicherheit für Videodaten," *Vieweg*, 1998.
6. L. Qiao and K. Nahrstedt, "Comparison of MPEG Encryption Algorithms," *Computers and Graphics*, vol. 22, no. 4, Aug. 1998, pp. 437-448.
7. L. Qiao and K. Nahrstedt, "Watermarking Methods for MPEG Encoded Video towards Resolving Rightful Ownership," *IEEE Multimedia Computing and Systems Conf.*, IEEE CS Press, Los Alamitos, Calif., 1998, pp. 276-286
8. J. Dittmann, "Digitale Wasserzeichen,"Grundlagen, Verfahren, Anwendungsgebiete, Springer, Berlin, 2000.
9. C.-Y. Lin and S.-F. Chang, "A Robust Image Authentication Method Surviving JPEG Lossy Compression," *SPIE Storage and Retrieval for Image and Video Databases*, SPIE Press, Bellingham, Wash., 1998, http://smart.iis.sinica.edu.tw/~lcs/icme_497.htm.
10. *ISO/IEC 9796, 1991 Information Technology - Security Techniques - Digital Signature Scheme Giving Message Recovery*, Int'l Organization for Standarization, Geneva, 1991.
11. *ISO/IEC 14888, 1998 Information Technology - Security Techniques - Digital Signatures with Appendix*, Int'l Organization for Standarization, Geneva, 1998.
12. J. Fridrich, "Methods for Tamper Detection of Digital Images," *Proc. Workshop Multimedia and Security* (ACM Multimedia 99), J. Dittmann, K. Nahrstedt, and P. Wohlmacher, eds., GMD, Bonn, Germany, 1999, pp. 29-34.
13. *ISO/IEC 9798, Information Technology - Security Techniques - Entity Authentication. Part 1, General. Part 2, Mechanisms Using Encipherment Algorithms. Part 3, Mechanisms Using a Public-Key Algorithm*, Int'l Organization for Standarization, Geneva, 1997.
14. *ISO/IEC 13888, Information Technology - Security Techniques - Non-repudiation. Part 1, General. Part 2, Using Private-Key Techniques. Part 3, Using Public- Key Techniques*, Int'l Organization for Standarization, Geneva, 1997.
15. *ISO/IEC 9594-8, Information Technology - Open Systems Interconnection - The Directory, Authentication Framework, Tech. Corrigendum 1*, Int'l Organization for Standarization, Geneva, 2000.
16. *ITU-T Recommendation X.509 (1997 E), Information Technology - Open Systems Interconnection - The Directory, Authentication Framework, 6-1997*, Int'l Telecommunication Union-Telecommunications, 1997.
17. S. Micali, *Enhanced Certificate Revocation*, Massachusetts Inst. of Technology, Cambridge, Mass., tech. memo MIT/LCS/TM-542, 1995.
18. S. Even, O. Goldreich, and S. Micali, "On-Line/Off-Line Digital Signing," *Proc. CRYPTO 89*, Springer, New York, 1998, pp. 263-275.
19. M. Kutter and F. Petitcolas, "Fair Benchmark for Image Watermarking Systems," *Proc. SPIE Conf. Electronic Imaging 99*, vol. 3657, SPIE Press, Bellingham, Wash., 1999, pp. 226-239.
20. S. Micali, *Certificate Revocation System*, US patent 5.793.868, Patent and Trademark Office, Washington, D.C., 1998.
21. R. Merkle, *Secrecy, Authentication, and Public-Key Systems*, doctoral dissertation, Dept. Electrical Engineering, Stanford Univ., Stanford, Calif.,1979.
22. M. Myers et al., *X.509 Internet Public-Key Infrastructure Online Certificate Status Protocol - OCSP*, Internet Eng. Task Force, Jun. 1999, http://www.rfc-editor.org/rfc/rfc2560.txt.
23. N. Nikolaidis and I. Pitas, "Digital Image Watermarking - An Overview," *Proc. IEEE Multimedia Systems, Multimedia Computing, and Systems*, 1999, pp. 1-6, http://poseidon.csd.auth.gr/signatures/.
24. F. Petitcolas, R. Anderson, and M. Kuhn, "Attacks on Copyright Marking Systems," *Proc. Second Int'l Workshop on Information Hiding 98*, LNCS 1525, Springer, Berlin, 1998, pp. 219-239.

25. S. Craver et al., "Can Invisible Watermarks Resolve Rightful Ownerships?," *IBM Cyber J.*, July 1996, http://citeseer.nj.nec.com/craver96can.html.
26. J. Fridrich, "Applications of Data Hiding In Digital Images," *Tutorial for The ISPACS Conference*, 1998, http://citeseer.nj.nec.com/fridrich98application.html
27. T. Pun, "Watermark Attacks," DFG V3D2 Watermarking Workshop, http://www.lnt.de/~watermarking, 1999.
28. I. Cox and J.-P. Linnartz, "Public Watermarks and Resistance to Tampering," *Proc. IEEE Int'l Conf. Image Processing*, CD-ROM, IEEE Press, Piscataway, N.J., 1997.
29. J. Dittmann et al., "Interactive Watermarking Environments," *Proc. Int'l Conf. Multimedia Computing and Systems*, IEEE CS Press, Los Alamitos, Calif., 1998, pp. 286-294.
30. M.J.J. Maes, "Twin Peaks: The Histogram Attack to Fixed Depth Image Watermarks," *Proc. Workshop Information Hiding*, 1998. http://citeseer.nj.nec.com/maes98twin.html.
31. M. Holliman and N. Memon, "Counterfeiting Attack on Linear Watermarking Schemes," *Proc. Workshop Security Issues in Multimedia Systems* (IEEE Multimedia Systems Conf. 98), 1998.
32. D. Boneh and J. Shaw, "Collusion-Secure Fingerprinting for Digital Data," *Proc. of CRYPTO 95*, LNCS 963, Springer, Berlin, 1995, pp. 452-465.
33. J. Dittmann et al., "Combining Digital Watermarks and Collision Secure Fingerprints for Digital Images," *Proc. SPIE Conf. Electronic Imaging 99, Security and Watermarking of Multimedia Contents*, vol. 3657, 1999, pp. 171-182.
34. J. Dittmann, A. Steinmetz, and R. Steinmetz, "Content-Based Digital Signature for Motion Pictures Authentication and Content-Fragile Watermarking," *Proc. IEEE Multimedia Systems, Multimedia Computing, and Systems*, vol. 1, 1999, pp. 574-579.
35. L. Zhang et al., "RSVP: A new Resource ReSerVation Protocol," *IEEE Network*, Sept. 1993, pp. 8-18.
36. T.-L. Wu, S. Wu, and F. Gong, "Securing QoS, Threats to RSVP Messages and Their Countermeasures," *Proc. IEEE/IFIP Int'l Workshop Quality of Service* (IWQoS), IEEE Comm. Soc., 1999, pp. 62-64.
37. F. Baker, B. Lindell, and M. Talwar, *RSVP Cryptographic Authentication*, RFC 2747, 2000.
38. *Proc. ACM Multimedia 2000 Workshop*, ACM Press, New York, 2000, http://www.acm.org/sigs/sigmm/MM2000/ep/toc.html.
39. J. Dittmann, K. Nahrstedt, and P. Wohlmacher, "Approaches to Multimedia and Security," *Proc. IEEE Int'l Conf. Multimedia and Expo*, 2000, pp. 1275-1278.

Jana Dittmann is the head of the compentence center C4M at the Fraunhofer Institute Integrated Publication and Information Systems (FHG-IPSI, a former GMD institute). She specializes in the multimedia security field, and her research is focused on digital watermarking for copyright protection and content-based digital signatures for data authentication. She has a PhD in computer science from the Technical University of Darmstadt.

Petra Wohlmacher is an assistant to the electronic signatures section head at the Regulatory Authority for Telecommunications and Posts. She has a Dipl.-Math. degree in mathematics and informatics from the University of Darmstadt, Germany, and a PhD in mathematics from the University of Klagenfurt, Austria. She is a German Informatics Society member and sits on the leading committee of the Working Group 2.5.3, which is dealing with reliable IT systems.

Klara Nahrstedt is an associate professor in the Department of Computer Science at the University of Illinois, Urbana–Champaign. Her research interests include multimedia communication services, protocols, security, end-point architectures for multimedia, end-to-end quality of service, and resource management in multimedia networked systems. She is the coauthor of *Multimedia: Computing, Communications, and Applications,* Second Edition (Prentice Hall, 2002), and recipient of the Early National Science Foundation Career Award, the Junior Xerox Award, and the IEEE Communications Society Leonard Abraham Award for Research Achievements. She has a BA in mathematics; an MSc in numerical analysis from Humboldt University, Berlin; and a PhD from the University of Pennsylvania's Computer and Information Science Department. She is an ACM and IEEE Computer Society member.

Readers may contact Dittmann at FHG-IPSI, Dolivostrasse 15, 64293 Darmstadt, Germany; email jana.dittmann@ipsi.fraunhofer.de.