Securing the delivery of digital content over the Internet

by A. O. Waller, G. Jones, T. Whitley, J. Edwards, D. Kaleshi, A. Munro, B. MacFarlane, and A. Wood

This paper looks at the security issues that arise for the browsing, selection and delivery of digital content over the Internet. Particular emphasis is placed on the problems of finding security solutions for microtransactions (small items of content) and micropayments (low-value content), and the digital rights management issues concerning the protection of content after it has been delivered to consumers. The paper concludes with a description of the secure content delivery system developed during the SIBIS project, which addresses many of the issues raised.

1 Introduction

Currently, the purchasing of digital content over the Internet is in its infancy. The most widespread use of the Internet for purchasing is for items of relatively high-value content using the credit card system, mirroring the use of well established telephone mail order systems. However, the delivery of large items of high-value digital content represents only a small fraction of the potential services that could be supported by the Internet. An ideal content delivery system should be able to support efficient and secure browsing, selection and delivery of content from large sizes down to just a few bytes (microtransactions) and of high value down to potentially a fraction of a penny (micropayments). Such a system would allow, for example, individual charging for the delivery of stock market quotes, newspaper articles and Web pages, or even pay-per-use charging for video streams (such as a penny a minute). Being able to support the payment on delivery for such low-value items of content as a single Web page potentially allows a far greater range of people and organisations to become content providers. To make this possible, however, the content delivery system must have low technical, financial and legal barriers to becoming a content provider.

In this paper we will give an overview of the current state of the art in secure content delivery systems for the Internet, focusing on the issues identified above. Section 2 gives an overview of the potential requirements for a general-purpose content delivery system for the Internet. In Section 3 we look at systems designed to protect the content during delivery, and analyse how well they meet the requirements of Section 2. These systems, however, do not protect the content after it has been delivered to the consumer. Recent high-profile cases, such as Napster[1], have highlighted the need to protect content against illegal copying and other licence infringements. Digital rights management (DRM) systems that aim to protect content against such licence rule infringements are looked at in Section 4. In Section 5 we briefly cover micropayment systems, and finally in Section 6 we describe the SIBIS (Secure Interactive Broadcast Infotainment Services) secure content delivery system, which has been developed as part of a two-year DTI/EPSRC Link Broadcast project.

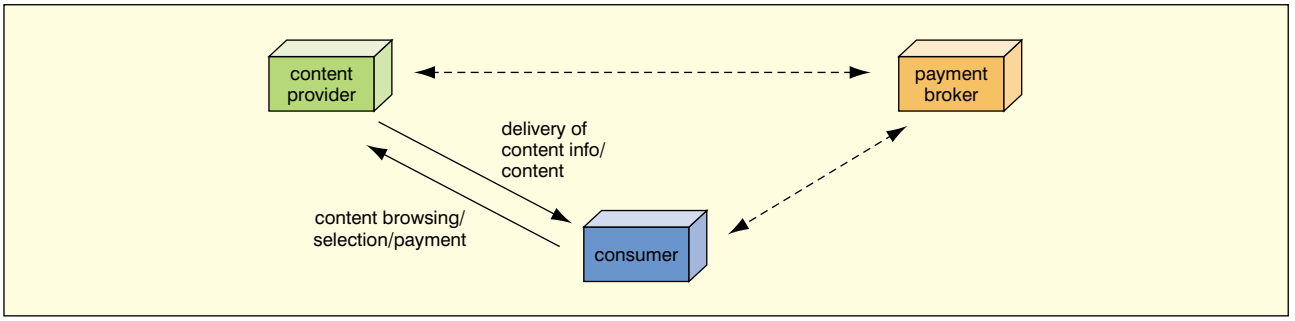

Fig. 1 Overview of a content delivery system

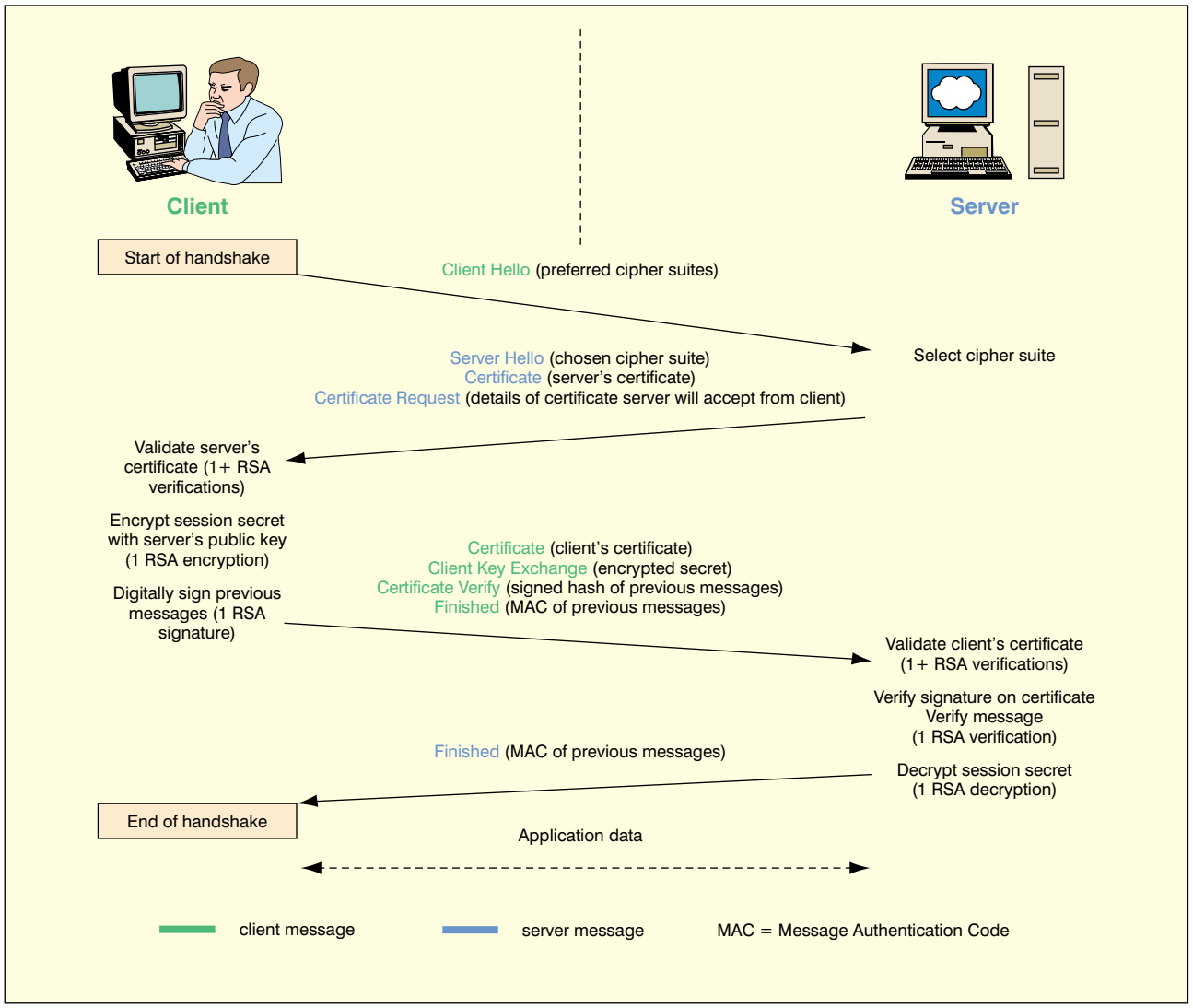

Fig. 2 A typical TLS handshake with client authentication

2 Requirements

The participants in a content delivery system, and the high-level interactions between them, are shown in Fig. 1. In general, consumers will perform some form of browsing followed by content selection and possibly payment to the content provider. The content provider will then deliver the content to the consumer (note that the producers and distributors of content are in general different but have been combined here into one entity for clarity). To handle the payments, there is typically a third-party payment broker, which may interact online or offline with the content provider and consumer. For example, with the credit card system the payment broker interacts with the consumer offline to set up and manage their account and obtain monthly payments, and interacts with the content provider online to authorise the payment for each purchase.

Some of the potential requirements of each of the participants are listed below.

- *The consumer* seeks guarantees of the integrity of content, ease of use and low cost of the system, protection from payment fraud, privacy protection for their personal and payment details and anonymity in their dealings with content providers (which can help in achieving privacy).
- *The content provider:* Content providers are primarily concerned with receiving payment for use of the content and hence preventing those who haven't paid for it from using it. This implies requirements for the confidentiality of content during content delivery to prevent eavesdroppers from obtaining content, authentication of consumers and receipt of payment from them before they can use the content and preferably protection of the content after it has been delivered to the consumer (DRM).
- *The payment broker* doesn't have requirements from the system as such, other than the ability to charge for the services that it provides.

Ideally, the content delivery system should be suitable for a wide range of scenarios to make it as flexible as possible. The following three requirements are crucial to achieve this flexibility.

- *Scalability:* The system should be able to support thousands of content providers and millions of

consumers. Ideally, almost anyone should be able to be a content provider, and for this to be possible the barriers in terms of costs of licences and hardware and the necessary trust required to be a content provider should be as low as possible.

- *Granularity of content size:* This refers to the delivery of content from just a few bytes (microtransactions) up to potentially gigabytes of data. An example of a microtransaction could be the purchase of a stock market quote or foreign exchange rate. Being able to support large volumes of microtransactions in an efficient way is a particular challenge.
- *Granularity of payment:* Support for payment from a fraction of a penny (micropayments) up to tens of pounds or even more should be possible. The ability to support micropayments would allow charging for Internet search results, for individual Web pages or for individual articles in an online magazine, for example.

In the remainder of this paper we will look at existing security systems to determine the degree to which they meet these requirements.

3 Securing the content delivery

Technologies designed to protect the delivery of content can be divided into two types: those that set up a secure connection down which to send the content ('secure the pipe') and those that apply security to the content directly ('secure the content').

'Secure the pipe'

As just mentioned, this type of content delivery security is characterised by the setting up of a secure session between the content provider and the consumer, which can then be used to transmit the content securely. The most commonly used technology of this type is SSL/TLS[2] (Secure Sockets Layer/Transport Layer Security), which is widely supported in Web browsers and Web servers. We will use TLS to illustrate the potential advantages and disadvantages of this type of content delivery.

The process involved in setting up a secure TLS session, the 'handshake', is illustrated in Fig. 2. This diagram shows the flow of messages between a client and server in the handshake, as well as the computationally intensive public-key-type operations (in this case RSA [Rivest, Shamir, Adleman] sign/verify or encrypt/decrypt operations) that the client and server have to do. Other operations have been omitted for clarity. As can be seen, this handshake involves several round-trip communications and public key operations. It may also involve the client and server interacting with a PKI (Public Key Infrastructure) to obtain certificate status information in order to validate each other's certificate. However, once the session has been set up, TLS uses symmetric key encryption and integrity protection techniques and can therefore provide protection to data in a relatively efficient way.

In terms of achieving the potential requirements identified in Section 2, TLS has many advantages. For the content delivery itself, it will protect the confidentiality and integrity of the content and will do this in an efficient way once the session has been set up. The secure session can also be used to protect the privacy of consumers by protecting the content browsing, selection and payment communications from potential eavesdropping. From the content providers' point of view the secure session can also provide them with authentication of the consumer's identity.

The major drawback of TLS is a potential lack of scalability, particularly for the scenario of micro-transactions where the session set up is a significant overhead. Handling a large number of connections on a TLS server within a short period of time is a problem that can require special hardware. In the case of micro-transactions, all of this effort has to take place just to transfer a small amount of content, which is not very efficient and may even be uneconomic for very low-value content. Another potential drawback is that the protection has to be applied at the same time as the content delivery. If the content provider is simultaneously handling a very large number of consumers, or if high-data-rate content delivery is required (for real-time protocols for example) then this represents a significant computational load that again can require special hardware at the server to enable it to cope. Finally, as the key used to decrypt the content is essentially the key of the secure session, the payment and authorisation decisions have to be part of the content delivery decision and therefore the content provider has to be involved in this. This has implications for consumer privacy, as consumers will in general have to provide sensitive information, such as payment details (e.g. credit card numbers), to the content provider. It can also place a barrier on becoming a content provider, as content providers may need to be relatively highly trusted.

In general, therefore, this type of content delivery security is most suited to the scenario of a consumer obtaining content in a session, where they are either obtaining a large volume of content or are selecting a number of different items of content from the same content provider.

'Secure the content'

With this type of content delivery, protection is applied directly to content, and therefore a secure delivery channel between the content provider and the consumer is not required. Typically the content is encrypted and wrapped into some kind of container, with appropriate header information added, and integrity protection is then applied by, for example, digitally signing the container. This container can then be delivered to any number of consumers without further processing being required by the content provider. The remaining problem is how to distribute the decryption keys to the consumers who are authorised to use the content (i.e. to those who have paid for it).

One way to solve this problem is to make use of a third-party transaction broker, and this is the approach used by digital container systems. Commercial systems usually integrate this approach with DRM, which we will look at more closely in Section 4.

The basic digital containers architecture is illustrated in

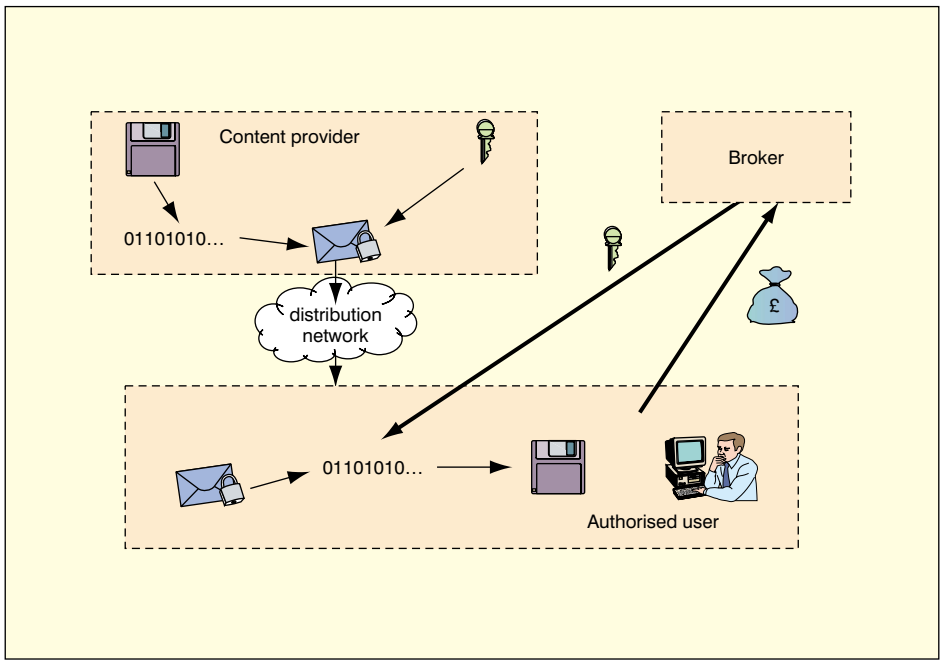

Fig. 3 Digital containers architecture

Fig. 3. As just mentioned, digital content (in this case represented by an arbitrary file) is wrapped into a protected container using a secret key, and this container can then be delivered to any number of consumers over any, possibly insecure, distribution network without compromising the security of the content. When a consumer wishes to use the content, they have to contact the broker with the appropriate authorisation details, such as a valid payment. The broker will then return the decryption key allowing the consumer to use the content, and will also handle the clearing of the payment with the content provider. The fine details of how the broker can generate a decryption key and the interaction between content providers and the brokers differ from system to system.

This approach has many advantages, particularly in terms of scalability. Protection can be applied once, offline, for any number of consumers, which means that potentially expensive cryptographic operations do not need to take place in real time at the content provider. This also protects the content while it is in storage. For these reasons, digital containers are particularly suited to store-and-forward content delivery systems. The content may also be delivered through any distribution channel, which means that even though usually it would be distributed directly from the content provider via the Internet, it could also be distributed on CDs for example. Another advantage is that it separates the content payment and authorisation from the content delivery, which frees content providers from having to handle payment and authorisation decisions and allows them to concentrate on content delivery. This freedom significantly reduces the amount of trust required between the broker and the content provider, thus reducing the content providers' costs and at the same time allowing a far larger number of people and organisations to be content providers. Furthermore, this offers an advantage from the point of view of the consumer, in that the content provider is not responsible for handling potentially sensitive payment and authorisation details of the consumers, and indeed consumers can be completely anonymous to the content providers with this approach.

For these reasons, digital containers are particularly suited to store-and-forward delivery of discrete items of content. Real-time content delivery, such as video streaming from point-to-point or even multicasting applications, could also be supported by dividing up the stream into suitably sized chunks for placing into containers. However, this 'chunking' approach could place a significant load on the broker in terms of the number of key requests it has to handle.

4 Digital rights management (DRM)

The technologies discussed in Section 3 are concerned with protecting the content while it is being delivered, but not after it has been delivered. DRM aims to address the protection of content throughout its lifetime. This leads to requirements such as copyright protection, prevention/detection of illegal copying and enforcing of other licensing rules (e.g. expiry of content, limited number of viewings etc.). Developing a secure content delivery system that incorporates DRM is attracting a lot of interest at the present time, and there are a large number of companies and organisations working on developing such systems. Major companies include Microsoft, IBM, RealNetworks and Sony, to name just a few, who are working on both the component technologies of the systems as well as providing completely integrated secure content delivery systems. Organisations include the MPAA (Motion Picture Association of America) and the SDMI (Secure Digital Music Initiative), who have been looking at developing systems for particular applications such as DVD copy protection and the secure delivery of digital music. In terms of the DRM technologies used in these proposed systems, there are two major types at present. These are digital watermarking and digital containers, and each of these is discussed below.

Digital watermarking

Digital watermarking is a technique that enables information to be embedded within digital content. This information could be, for example, the copyright holder's identity or licence rules that apply to the content. For a good introduction to digital watermarking techniques and how they can be applied, see the paper by Podilchuk and

Delp[3]. Example applications include:

- keeping track of content to detect illegal copying
- controlling, in conjunction with special hardware, the number of copies that can be made by embedding copy control information
- detecting modifications to digital content by the use of 'fragile' watermarks, which any such modifications will destroy
- monitoring broadcast services to, for example, track the number of uses of adverts or a song for royalty and other payment purposes.

For our scenario of thousands of content providers wishing to protect digital content after it has been delivered to consumers, the main application of watermarking would be in embedding information within content to allow illegal copying and distribution to be detected. For this application, watermarks need to be 'robust' in that it should not be possible for consumers to remove the watermark without harming the quality of the content so much that it becomes worthless. This implies that the watermark must be embedded throughout the content, otherwise just the part that contained the watermark could be removed. There are two kinds of robust watermarks: visible and invisible.

Fig. 4 A scanned page from the Vatican library showing a visible watermark using IBM's Digital Library software

Visible watermarks can be used to embed copyright information within images. An example of a visible watermark is shown in Fig. 4. Clearly, visible watermarks are only of limited use due to their effect on the content. In the above example, the watermark does not stop researchers from being able to read the words on the page and hence the content remains useful, however it would not be acceptable for there to be a similar watermark visible throughout a Hollywood movie.

Invisible watermarks aim to embed information within redundancy in the content in such a way that the watermarked content is perceptually the same as the original. As the watermarks are invisible, this helps with robustness as attackers will not know where the watermark is embedded. It does mean, however, that a secret key must be used to embed the watermark and allow its later extraction by the content provider. Techniques to do this exist for most content types, including pictures, audio and video.

An example of a potential use of invisible watermarking is fingerprinting, which is illustrated in Fig. 5. With fingerprinting, when a consumer purchases some digital content, the content provider embeds a watermark containing the consumer's identity. If at a later stage that consumer decides to illegally distribute the content, and if the content provider obtains a copy, then they are able to extract the consumer's identity from the watermark and know who was responsible for the illegal distribution. As the watermark is invisible, it does not affect in any way the consumer's use of the content. In fact, they will in general not be aware that there is a watermark in the content, although for this application the fact that their identity is embedded as a watermark would be likely to be made aware to them to act as a deterrent against illegal copying and distribution.

An important advantage of using robust, invisible watermarking for DRM is that the watermarks are transparent to the consumers and to the software in existing content delivery systems. They can therefore be used with minimal impact on any existing system. However, there are doubts about how robust they actually are when subject to a deliberate attack. A good example of this is the SDMI (Secure Digital Music Initiative) challenge. The SDMI is a consortium of music industry companies who are working to develop and standardise technologies to protect digital music. In September 2000 they issued a challenge to try and remove the robust, invisible watermarks on audio content they had created using four different watermarking schemes (including the Verance scheme used to protect DVD audio). Despite the conditions of the challenge being very restrictive, with only three weeks allowed and no details of the algorithms used being provided, a team from Princeton and Rice universities managed to defeat all four watermarking schemes[4]. Work on trying to improve the robustness of watermarking schemes is however taking place. An important initiative in this area may be the production of benchmarking tools, such as Checkmark[5] and the work of the Certimark[6] project, which combine a large variety of the previously known and new attacks. The aim of these tools is to assess the robustness of watermarking schemes and their suitability for particular applications, and hopefully to lead to improvements in new watermarking schemes.

Digital containers

DRM technologies based on digital containers combine 'secure the content' content delivery with the use of

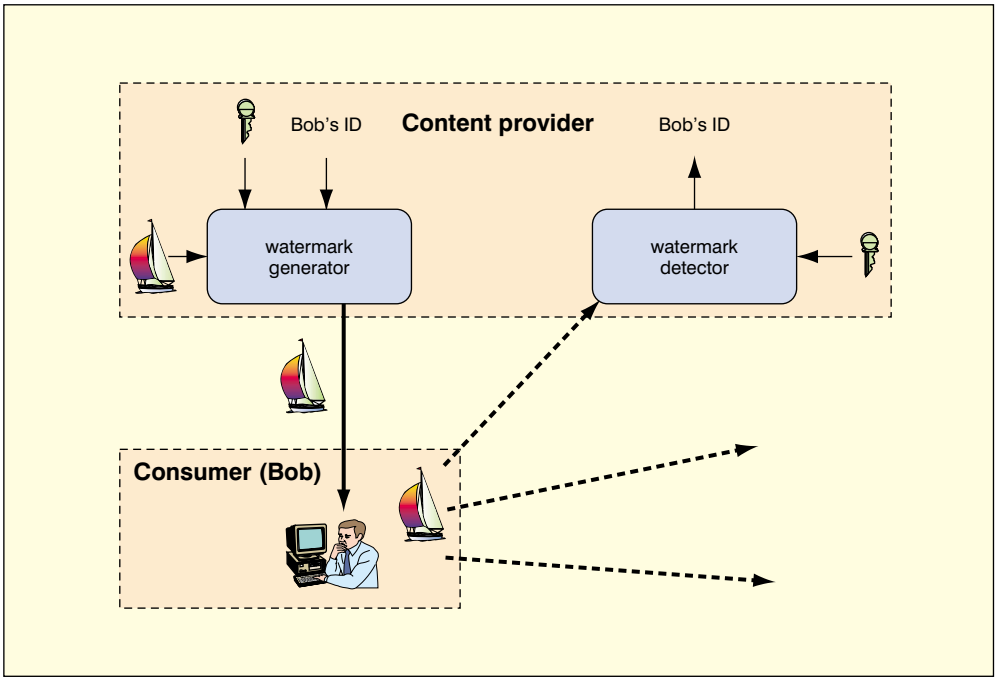

Fig. 5 Fingerprinting using robust, invisible watermarks

'tamper resistant' software or hardware at the consumer. This software is responsible for handling the decryption and rendering of the content and must prevent the consumer from gaining access either to the decryption key or to the decrypted digital content. This is illustrated in Fig. 6.

Commercial examples of such systems include IBM's Electronic Media Management System[7] and Microsoft's Windows Media Rights Manager[8]. For a comparison of some of the more prominent schemes see the White Paper by Sonera Plaza Ltd Medialab[9].

This approach inherits all of the advantages of the digital containers content delivery architecture in terms of scalability, and for DRM it should be able to prevent copyright or licence fraud, not just detect it. It also enables the enforcement of new licensing rules such as pay per use or expiry of content as these can be enforced by the tamper-resistant software or hardware. A final advantage is that so-called 'superdistribution' (the idea that consumers themselves, or indeed anyone, can become distributors of content without any loss of security) is supported, which can reduce the burden on the original content provider and enhance scalability. As an example, consumers who like a particular item of content may wish to distribute this to their friends. They are prevented from accessing the digital content itself by the tamper-resistant software or hardware, but they can still distribute the protected digital containers. Anyone who receives these will still have to pay for the content to be able to use it.

There are some potential problems however. Perhaps the most serious issue, as with digital watermarking, is doubt about the security of this technique, particularly if implemented in software. How 'tamper resistant' is the software? An example of digital containers software that has recently been circumvented, and which was widely reported in the press[10], is version 2 of Microsoft's DRM software. This particular attack also illustrated the problem that it only takes one person (in this case 'Beale Screamer') to work out how to get around the system and they can then distribute software to automate the attack, which allows anyone else to get around the system. Another potential problem is that, from a practical point of view, this technique requires special software or hardware for each type of digital content as the content decryption has to be tightly coupled to the rendering application. This is to prevent consumers gaining access to the decrypted digital content. Existing rendering applications would therefore have to be modified to support digital container systems and this could also lead to consumers having to have multiple renderers for the same type of content. This situation could be eased by standardisation of digital container technologies, or perhaps more likely by the consolidation of the market to only a handful of competing systems. In any case, consumers already have to have several different renderers for the same type of content at present, such as for digital audio, so this may not be a significant problem.

5 Micropayments

For micropayments to be viable, the intrinsic costs of the content delivery system must be evaluated in terms of computation, communications and equipment required. As previously mentioned in Section 3, 'secure the pipe' delivery can suffer from significant overheads at session set-up, which mean that these system costs are likely to be relatively high. Therefore, 'secure the content' delivery is likely to be more suited to the support of micropayments.

In terms of micropayment systems themselves, current support is dominated by indirect payment systems where content is paid for by means other than direct purchase at the time of the content retrieval. This covers advertising and subscription. Advertising can meet many of the requirements, however it is not suitable for the support of a large number of small content providers and would need

to be supplemented by another payment system for high-value content. It also provides a barrier to starting up as a content provider. Subscription can also meet many of the requirements, however it is not suited to the support of infrequent microtransactions or where consumers may wish to use a large number of different content providers.

A more general solution would therefore need a direct micropayment system. In recent years, there has been a considerable amount of research into developing such systems, although few if any have gained widespread support. It is beyond the scope of this paper to cover these systems in detail, however a good source of information on proposed micropayment systems is the 'ePayment Systems Observatory'[11], which has a comprehensive list, as well as a description, of a large number of them.

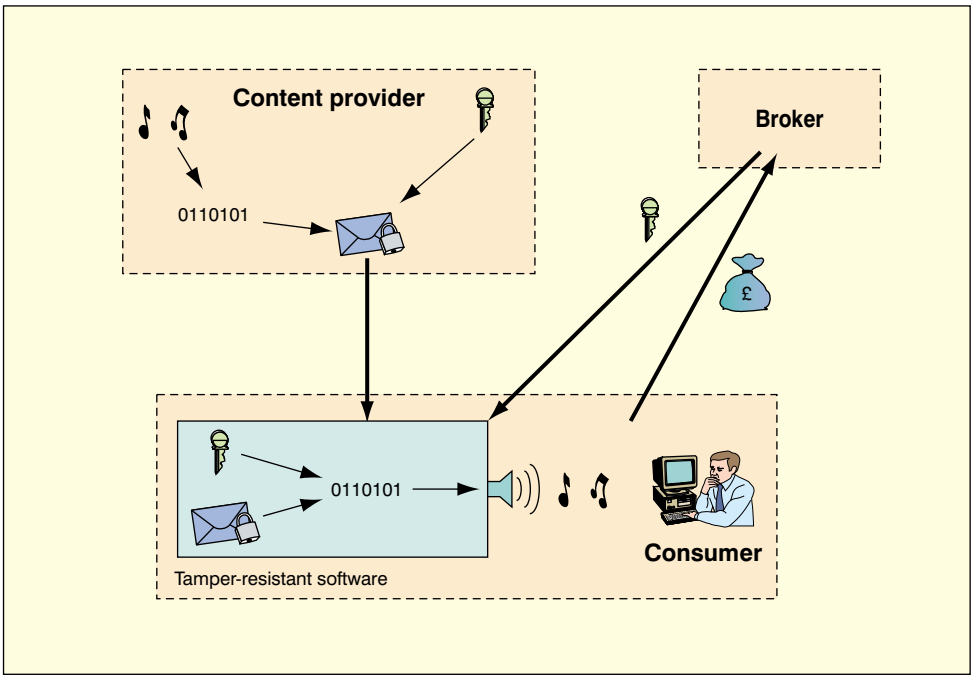

Fig. 6 Digital containers with DRM

Micropayment systems can be divided into two types. Token systems use some kind of marker representing a value as the medium of exchange. A physical example of this would be cash. Notational systems exchange a value by authorisation, such as with a typical bank account. While it is true that either type of system could meet the requirements for a secure content delivery system, the notational systems can provide a particularly good fit with 'secure the content' delivery systems. This is because a broker has to be contacted by the consumer to obtain the decryption key. If at the same time the broker handles the billing of the consumer for the content, such as by transferring money from the consumer's to the content provider's account, micropayments can be handled in a particularly efficient way. This adds relatively little in the way of overhead to the content delivery system in order to support payments.

6 The SIBIS system

The SIBIS (Secure Interactive Broadcast Infotainment Services) project is a DTI Link Broadcast project with the following aim:

> 'To develop the protocols, methods, and functionality to enable scalable, secure, and reliable electronic microtransactions to take place between consumers and content providers'.

The original project partners were Thales Research and Technology (UK), University of Bristol (Department of Electrical and Electronic Engineering and Department of Computer Science), Pedagog and Zygo Communications. During the project we have produced a specification of a content delivery system. To achieve the aim of being able to support microtransactions and of being able to scale to a large number of content providers and consumers, we decided to base the system on the digital containers architecture described in Section 3. This architecture is particularly suited to microtransactions due to the relatively low overhead of content delivery. It is also particularly suited to micropayments as payment occurs within the brokers and can therefore use efficient account-based billing. Due to the separation of payment and authorisation from content delivery, the brokers in the SIBIS system can also treat content providers as little more than special consumers, thus opening up the possibility of almost anyone being a content provider.

The SIBIS system does not, however, include provisions for DRM after the content has been paid for. There were a couple of reasons for this. Firstly, the system is specifically designed for microtransactions and micropayments where the value of content is relatively low and hence the copying and redistribution of content is likely to be a relatively minor concern. Having said that, the system may be suitable for delivery of high-value content that is only short lived, such as stock market quotes or certain kinds of live event (e.g. a football match). Secondly, we decided that the maturity of the rights management technologies was not sufficient to provide enough guarantee of the security of the content to make their incorporation worthwhile. This may change in the future, and if so either digital watermarking or the rights management facilities of digital containers technologies would be relatively easy to integrate into the SIBIS system.

Some details of the components in the SIBIS system, as well as the protocols and data formats specified by SIBIS,

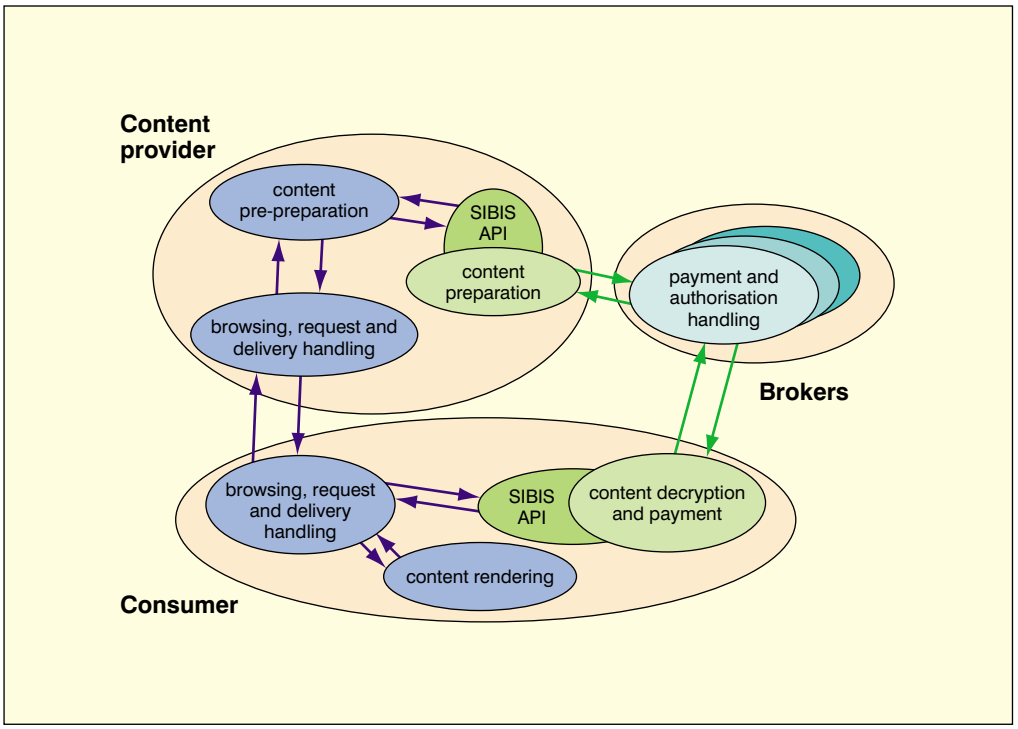

Fig. 7 Component view of the SIBIS system

are given in the following subsections. Note that the final part of the project will involve the production of a demonstrator and the execution of medium-scale trials with it at the University of Bristol in order to validate the performance of our design.

The SIBIS components

A component view of the SIBIS system, including the SIBIS API (Application Program Interface), is shown in Fig. 7. These APIs provide the interface to SIBIS components that handle the packaging of content into digital containers at the content provider and the extraction of content from the containers at the consumer, and are designed to allow seamless integration with existing content delivery systems. Note that this approach means that the SIBIS system is independent of the content delivery channel, and can therefore support content delivery channels other than the Internet.

The content prepreparation stage at the content provider divides content up into suitably sized chunks to be placed in containers. The chunk sizes are determined based on the type of content being requested, the characteristics of the delivery channel, and the payment method. In many cases this will simply provide the content as it is to the SIBIS API, but could be used for example to divide up a stream of data, such as a video stream, into chunks to allow consumers to be charged only for the particular parts of the stream that they actually use. In either case, the rest of the content delivery should be unaffected.

At the consumer, the content delivery handling software (e.g. a Web browser) will invoke the consumer SIBIS API to handle the SIBIS content (i.e. exchange the payment with the broker for the keys to the digital containers), and deliver the decrypted content to the rendering software.

The SIBIS specification

The SIBIS specification defines the interaction between a content provider and its broker, the interaction between a consumer and its broker and the construction and format of the digital containers. The main considerations when designing this specification were to make sure that the system would be secure and scalable. In terms of security, the SIBIS participants need to be protected not only from the usual external attackers, who may wish to eavesdrop on communications for example, but also from the other SIBIS participants. Clearly, consumers cannot in general be trusted to behave correctly and may attempt to abuse the system for their own gain if this is at all possible. Content providers may also do this, particularly with our aim that almost anyone can be a content provider. Measures were therefore included within the SIBIS specification to prevent the participants gaining from acting maliciously within the system.

It is not within the scope of this paper to provide a detailed description of the SIBIS specification, however a summary of some of the important features is given below.

Content provider-broker interactions: The interaction between a content provider and its broker is limited to two areas. Firstly, in order to receive payments for content the content provider needs to open and maintain an account with its broker. Secondly, to allow its broker to regenerate content encryption keys (to be passed to consumers) the content provider needs to share a 'master' secret with its broker that will be used to derive these keys. These interactions mostly take place at set-up time and can be performed off-line.

SIBIS containers: To create a SIBIS digital container, the content provider must first generate a content encryption key. This key is derived using the 'master' secret, a unique content ID (to make the key specific to that item of

content) plus certain other information—including the price of the content—whose integrity needs to be protected from consumers. The purpose of including the latter information is to protect content providers from consumers who attempt to mislead brokers. Consumers may, for example, attempt to tamper with the price of the content so that they are charged less for it. Including the price of the content in the derivation of the key prevents this from happening since the result of changing the price would be the generation of the wrong key.

Once the key has been derived it is used to encrypt the content. The details used to derive the key, excluding the 'master' secret itself of course, are then placed in the container together with the encrypted content. Other information that may be useful to consumers, such as a description of the content, is also included and, finally, the container is digitally signed by the content provider. This digital signature enables consumers to check the integrity of the content. In the case that the content provider accidentally or maliciously makes errors in containers, the signature also provides evidence that consumers can use to prove wrongdoing by the content provider.

The precise format of a SIBIS container is quite complicated, however a simplified view of its contents is given in Fig. 8.

Consumer–broker interactions: When a container is received by a consumer, the consumer needs to obtain the decryption key from its broker. In the SIBIS system, a secure connection is established using SSL/TLS between the consumer and its broker for this purpose. The secure connection provides authentication of the consumer and broker to each other, as well as privacy and integrity protection for their communications. As mentioned in Section 3, there is a significant overhead when this connection is established. However, it need only be established once at the start of the consumer's content purchasing session and can remain open until the end of this session. Therefore, the overhead can in general be spread over key requests for many items of content.

To request a decryption key for the content, the consumer sends to its broker the key derivation details, which include the price of the content, obtained from the container. Note that it is also possible for the consumer to batch requests for keys and therefore obtain the keys for several containers at the same time to enhance scalability. It is also important to note that the key derivation details only contain a content ID and no other description of the content. If this ID, as would be the case in general, is simply a random number, then the consumer's privacy is protected as the broker will not know what content the consumer is purchasing.

On receipt of the consumer's request, the broker regenerates the key and returns it to the consumer. This step will in general require the consumer's broker to forward the request details to the content provider's broker and obtain the key from it. On receipt of the key, it is possible that the consumer will fail to decrypt the content for some reason, such as the key being derived incorrectly. In order to protect consumers from being charged for content that they cannot use, charging will not take place until a confirmation message is sent from the consumer to its broker that the decryption was successful. Clearly, this provides an opportunity for consumers to claim that they have not successfully decrypted content and thus enable them to obtain content for free. This will need to be dealt with by means outside of the SIBIS specification, such as monitoring consumers for abnormal behaviour and investigating persistent offenders in more detail.

7 Conclusion

We have seen that in order to support microtransactions and micropayments the digital containers architecture offers many advantages. By combining this architecture with the use of tamper-resistant software or hardware at the consumer or with digital watermarking, digital rights management for the lifetime of the content can also be achieved. However, further work is likely to be required on improving and demonstrating the security of these rights management systems, as well as investigating the applicability of this architecture to real-time content delivery.

Header	Version information
	Content description—Includes the MIME type of the encapsulated digital content as well as a user readable description.
	Key derivation details—Includes information needed by the broker to regenerate the decryption key, including the content ID, price, content provider ID etc.
	Sequence number (optional)—This can be included to identify a container if it is part of a sequence, such as a video stream.
Content	Encrypted content—The encrypted digital content.
Integrity protection	Digital signature—This is calculated on the previous fields and protects both the header and the content.
	Certificates (optional)—This contains the content provider's digital certificate chain, which is needed by consumers to validate the signature on the container. Due to the potentially large size of these, they may be distributed by out-of-band means instead.

Fig. 8 Simplified view of a SIBIS container

Adrian Waller gained a BSc in Mathematics from the University of Surrey in 1992 and a PhD in Pure Mathematics from Royal Holloway, University of London in 1996. After a year as a researcher at the University of Ljubljana, Slovenia, he joined TRT(UK) in 1997. He is currently a Specialist Principal Engineer providing information security expertise to projects within the Internet Technology Centre at TRT (UK).

Address: Thales Research and Technology (UK) Ltd., Worton Drive, Worton Grange, Reading RG2 0SB, UK
E-mail: adrian.waller@uk.thalesgroup.com

Glyn Jones is a Chief Engineer with Thales Research and Technology (UK) (formerly Racal Research), where he leads communications and networking projects. He moved to TRT(UK) in 1997 from GEC Hirst Research Centre, where he worked on telecommunications switching and multiplexing. Before 1986 he was employed by Thorn-EMI. Glyn gained a BSc Honours degree in Applied Physics and Electronics from Durham University in 1978.

Address: Thales Research and Technology (UK) Ltd., Worton Drive, Worton Grange, Reading RG2 0SB, UK
E-mail: glyn.jones@uk.thalesgroup.com

Toby Whitley studied for BEng and PhD degrees at Bristol University. His PhD was on Exploring Mobility in Wireless ATM Networks, looking particularly at broadband and multiservice handover. He has also worked on the Tsunami II project on calibration of beamforming hardware and also done some work on antenna design for radar and ground-penetrating radar for the DERA. He is now taking time off to cycle down the east coast of Australia.

E-mail: toby_whitley@yahoo.co.uk

James Edwards gained a BSc in Computer Science (Bristol, 2000) and is currently a Research Assistant and PhD student in the System Verification Group at the University of Bristol. His current research interests include the formal modelling and analysis of cryptographic systems and he has considerable experience in process algebras and software engineering.

Address: Room 3.39, Merchant Venturers Building, University of Bristol, Woodland Road, Bristol BS8 1UB, UK
E-mail: edwards@cs.bris.ac.uk

Dritan Kaleshi received a degree (Excellent Diploma) in Electronic Engineering from the Polytechnic University of Tirana, Albania, in 1991. He was a lecturer there until 1996, when he joined the Centre for Communications Research at University of Bristol, UK. He has participated in several EU and UK funded research projects (ETHOS, SCARF, M-VCE, SIBIS). His main research interests cover self-configurable distributed systems, wireless networking and home-area networking technologies. He is currently a Lecturer in Communications Networks with the Department of Electrical and Electronic Engineering, University of Bristol, UK.

Address: Room 5.06, Merchant Venturers Building, University of Bristol, Woodland Road, Bristol BS8 1UB, UK
E-mail: dritan.kaleshi@bristol.ac.uk.

Alistair T. Munro received a BSc degree from Imperial College, London, in 1975 and a PhD degree from UMIST, Manchester, in 1983. While a Reader in the Department of Electrical and Electronic Engineering at the University of Bristol his research was concerned with distributed processing systems: their architecture and design (with emphasis on mobility), the algorithms and protocols they execute; their performance (do they work? how well do they work?); and their realisation and deployment. He now works in industry and is a Visiting Professor with the Networks and Protocols Group of the Centre for Communications Research, University of Bristol.

Address: Department of Electrical and Electronic Engineering, University of Bristol, Woodland Road, Bristol BS8 1UB, UK
E-mail: alistair.munro@bristol.ac.uk

Bruce MacFarlane is Technical Director and co-founder of Pedagog Ltd., one of the UK's leading developers of wireless multimedia applications. A winner of the European IT Award, he has extensive experience in the application of technologies such as neural networks and video image processing to real-world solutions.

Address: Pedagog Ltd., Tredomen Business Centre, Hengoed CF82 7FN, UK
E-mail: bruce@pedagog.com.

Angus Wood is an independent technology consultant in the fields of distributed systems R&D and security. He is currently engaged as a Wireless Systems Consultant by Inspired Broadcast Networks in London, UK.

E-mail: gus@shady.org

Acknowledgments

The authors gratefully acknowledge the financial support provided by the DTI/EPSRC LINK Broadcast project 'Secure Interactive Broadcast Infotainment Services (SIBIS)' (No. TJBF/C/002/00034) for the work on which this article is based, and Thales Research and Technology (UK) for allowing its publication.

References

1 http://news.bbc.co.uk/hi/english/entertainment/new_media/newsid_1586000/1586226.stm
2 DIERKS, D., and ALLEN, C.: 'The TLS Protocol'. RFC 2246, IETF Network Working Group, January 1999
3 PODILCHUK, C. I., and DELP, E. J.: 'Digital watermarking: algorithms and applications', *IEEE Signal Process. Mag.*, July 2001, **18,** (4), pp.33–46
4 CRAVER, S. A., DEAN, R. D., FELTEN, E. W., LIU, B., STUBBLEFIELD, A., SWARTZLANDER, B., WALLACH, D. S., and WU, M.: 'Reading between the lines: lessons from the SDMI challenge'. Proc. 10th USENIX Security Symposium, 13th–17th August 2001, Washington, DC
5 PEREIRA, S., VOLOSHYNOVSKIY, S., MADUEÑO, M., MARCHAND-MAILLET, S., and PUN, T.: 'Second generation benchmarking and application oriented evaluation'. Information Hiding Workshop III, April 2001, Pittsburgh, PA, USA (Springer Lecture Notes in Computer Science, 2001)
6 CERTIMARK: 'Certification for watermarking techniques', IST-1999-10987. See http://vision.unige.ch/certimark
7 See http://www.ibm.com/software/emms
8 MICROSOFT DIGITAL MEDIA DIVISION: 'Security overview of Windows Media Rights Manager', September 2001. See http://www.microsoft.com/windows/windowsmedia/WM7/DRM/whitepapers.asp
9 SONERA PLAZA LTD MEDIALAB: 'Digital rights management white paper', 3rd February 2002. See http://www.medialab.sonera.fi/workspace/DRMWhitePaper.pdf
10 See http://www.cnn.com/2001/TECH/internet/10/25/ ms.hacked.idg/
11 See http://epso.jrc.es

First received 20th June and in revised form 19th September 2002