End-to-end security in active networks

Ian Brown

a thesis submitted for the degree of
Doctor of Philosophy in Computer Science
University of London

Department of Computer Science
University College London
September 2001

Abstract

Active network solutions have been proposed to many of the problems caused by the increasing heterogeneity of the Internet. These systems allow nodes within the network to process data passing through in several ways.

Allowing code from various sources to run on routers introduces numerous security concerns that have been addressed by research into safe languages, restricted execution environments, and other related areas. But little attention has been paid to an even more critical question: the effect on end-to-end security of active flow manipulation.

This thesis first examines the threat model implicit in active networks. It develops a framework of security protocols in use at various layers of the networking stack, and their utility to multimedia transport and flow processing, and asks if it is reasonable to give active routers access to the plaintext of these flows. After considering the various security problems introduced, such as vulnerability to attacks on intermediaries or coercion, it concludes not.

We then ask if active network systems can be built that maintain end-to-end security without seriously degrading the functionality they provide. We describe the design and analysis of three such protocols: a distributed packet filtering system that can be used to adjust multimedia bandwidth requirements and defend against denial-of-service attacks; an efficient composition of link and transport-layer reliability mechanisms that increases the performance of TCP over lossy wireless links; and a distributed watermarking service that can efficiently deliver media flows marked with the identity of their recipients. In all three cases, similar functionality is provided to designs that do not maintain end-to-end security.

Finally, we reconsider traditional end-to-end arguments in both networking and security, and show that they have continuing importance for Internet design. Our watermarking work adds the concept of splitting trust throughout a network to that model; we suggest further applications of this idea.

Table of contents

Table of figures

Acknowledgements

This research would not have been possible without:

- my family, especially my mum and dad;
- the eternally patient Peter Kirstein and Steve Hailes;
- long and fruitful discussions in the Jeremy Bentham with Jon Crowcroft and Ken Carlberg;
- my other friends and colleagues: Anne Adams, Ross Anderson, John Andrews, Dan Armstrong, Adam Back, Dave Banisar, Simon Bell, Roy Bennett, Dave Bentley, Peter Bentley, Nicholas Bohm, Anna Bouch, Stuart Butchers, Caspar Bowden, Dominic Brady, Sacha Brostoff, Andrew Brown, James Brown, Stuart Butchers, Keith Campbell, Joc Chappell, Ben Crowe, Dan Crowe, Laura Dekker, Greg Fitzharris, Brian Gladman, Dean Griffiths, Ian Grigg, Tristan Henderson, Orion Hodson, Nadia Kausar, Jungwon Kim, Dave Lewis, Hugh Mallinson, Simon Martin, David Maxwell, Piers O'Hanlon, Tom Owen, Colin Perkins, Farez Abdul Rahman, Michael Rang, Elinor Rodgerson, Angela Sasse, Phil Treleaven, Matt Trotter, Tom Tuppen, Anna Watson, Carol Webb, Dirk Weirich and Gillian Wilson;
- EPSRC, UCL, Hidden Footprints, SAIC and the US government;

- And especially Simon Davies and Gus Hosein, who have shown that good tapas and rioja are the most important ingredients of any successful thesis.

I hope we have a lifetime of similarly enjoyable collaboration and friendship ahead.

Related publications

The following research has contributed to this thesis. It is organised here by the chapter describing the relevant work, and listed with citations and Web location.

Communications security

Ken Carlberg and Ian Brown. Framework for Supporting IEPS in IP Telephony. *IETF draft*, November 2000. http://www.cs.ucl.ac.uk/staff/K.Carlberg/saic/draft-carlberg-IEPS-Framework-00.txt

Ian Brown. Securing IEPS over IP. *ETSI TIPHON #22 document 58*, Bethesda, March 2001. http://docbox.etsi.org/tech-org/tiphon/Document/tiphon/05-200103-Bethesda/22TD058 -IEPS Security.doc

Secure multimedia conferencing

Peter Kirstein, Ian Brown and Edmund Whelan. Secure multicast conferencing. *Proc. DARPA Information Survivability Conference*, Hilton Head Island, South Carolina, January 2000. http://www.cs.ucl.ac.uk/staff/I.Brown/pimms/secure-conf.ps

Unconventional threats to active networks

Nicholas Bohm, Ian Brown and Brian Gladman. Strategic export controls: the impact on cryptography. *The Computer Law and Security Report*, 5(2), March/April 1999. http://www.fipr.org/publications/fexport.html

Ian Brown and Gus Hosein. Serve Yourself... Shifting Power Away from the Brothers. *Proc. ACM Computers, Freedom and Privacy 2000*, Toronto, April 2000. http://www.cs.ucl.ac.uk/staff/I.Brown/little-brother/

Ian Brown, Simon Davies and Gus Hosein (ed). The Economic Impact of the Regulation of Investigatory Powers Bill. *British Chambers of Commerce report*, June 2000. http://www.britishchambers.org.uk/newsandpolicy/ict/ripbillsummary.htm

Ian Brown. The Regulation of Investigatory Powers Act 2000. *International Forum on Surveillance by Design* (invited talk), London, September 2000.

Ross Anderson, Nicholas Bohm and Ian Brown. Maintaining consumer confidence in electronic payment mechanisms. *Cambridge Economic Crime Symposium* (invited talk), September 2000.

Nicholas Bohm, Ian Brown and Brian Gladman. Electronic commerce: who carries the risk of fraud? *Journal of Information, Law and Technology*, October 2000. http://elj.warwick.ac.uk/jilt/00-3/bohm.html

Ian Brown and Ben Laurie. Security against compelled disclosure. *Proc. 16th Annual Computer Security Applications Conference*, New Orleans, December 2000. http://www.acsac.org/2000/papers/47.pdf

Simon Davies and Ian Brown. Expert defence witness report for *Reg. vs. Connell*, Redbridge Youth Court, 19—20 October 2000.

Ian Brown. E-commerce and fraud. *Science, Technology and Justice* (invited talk), London, November 2001.

Distributed fingerprinting

Ian Brown, Colin Perkins and Jon Crowcroft. A method and apparatus for generating multiple watermarked copies of an information signal. *Patent Cooperation Treaty application WO00/56059*, March 1999. http://www.delphion.com/details?pn=WO00056059A1

Brian Gladman, Nicholas Bohm and Ian Brown. Could New Chip Privacy and Security Measures Tie Users' Hands? *Comment to Intel Corp.*, July 1999. http://www.cs.ucl.ac.uk/staff/I.Brown/chip-sec.htm

Ian Brown, Colin Perkins and Jon Crowcroft. Watercasting: distributed watermarking of multicast media. *Proc. Networked Group Communication '99*, Pisa, November 1999. http://www.cs.ucl.ac.uk/staff/I.Brown/pimms/watercast.ps.gz

Ian Brown and Simon Davies. The new corporate threat to freedom of expression. *Third UNESCO Congress on Ethical, Legal and Societal Challenges of Cyberspace*, Paris, November 2000. http://webworld.unesco.org/infoethics2000/report_151100.html#brown

Conclusion and further work

Ian Brown and Ian Grigg. Distributed Mailing Lists. *Work-in-progress*, November 1997. http://www.cs.ucl.ac.uk/staff/I.Brown/dml/dml.html

Ian Brown, Adam Back and Ben Laurie. Forward Secrecy Extensions for OpenPGP, *IETF draft*, August 2000. http://www.ietf.org/internet-drafts/draft-brown-pgp-pfs-01.txt

1 Introduction

"The nation's growing dependency on technology systems has resulted in a heightened vulnerability of our banking system, critical transportation networks, and vital government services, while also significantly increasing the incidence and complexity of crime."

–US Attorney-General John Ashcroft

"Imagine the disruption to the nation's infrastructure caused by someone's failure to auction off their great grandmother's curios on e-Bay."

–Wayne Madsen

"To date, we in the Senate have heard a great deal about the needs of law enforcement in the digital age and the risk that robust encryption poses to the traditional methods employed by law enforcement. At the same time, we have heard almost nothing about the privacy interests of law-abiding citizens."

– Senator John Ashcroft

The Internet is now beginning to fulfil its original promise as a universal communications medium. The level of traffic is currently doubling annually [Odlyzko01] and will continue to grow as connectivity is embedded into almost all electronic devices. Safety-critical systems such as medical infrastructure are being brought on-line.

The resulting importance of the network as a piece of "critical infrastructure" has already led to a raft of new legislation to regulate and police on-line activity [Brown00b]. This political energy, along with that spent making the Internet *insecure* for intelligence purposes over the last 25 years [Diffie97], might better have been spent on securing critical network services. But even network designers are developing systems that reduce rather than improve security.

This thesis is concerned with the security implications of a currently popular concept called active networking. Its aim is to transform the Internet into a massive computational resource that can be programmed and reconfigured by users [Tennenhouse97]. Applications can run on arbitrary nodes and even be part of packets, executing at each stage of their journey as they travel through the network. Programs move right down into the network and data-link layers. This allows much easier reconfiguration of the network and its protocols.

Active networks would require a radical re-engineering of the Internet. Amir *et al.* suggest instead that in-network computation be restricted to the application layer [Amir98]. By preserving the existing routing and forwarding semantics of the current Internet architecture, services can be deployed gradually in today's networks.

Many different types of active services have been proposed. They can be grouped into three types:

1. Gateways between networks using different protocols such as IPv4 and IPv6 or the Public Switched Telecommunications Network and the Internet [Brown01]. These gateways are becoming less useful as the world standardises on IP as a universal transport.
2. Distribution nodes that provide copies of large amounts of data to several clients in order to conserve bandwidth. These can work at the network or application layer, and provide real-time and delayed copies. Content Distribution Networks containing these nodes are a fast-growing industry sector. The nodes can alter the traffic passing through, filtering or transcoding data to reduce bandwidth requirements or provide individual versions of data to different clients.
3. Nodes that hide the source and destination of traffic by encrypting and reordering data passing through, such as anonymous remailers [Chaum81] and Onion Routing networks [Reed98b].

But whether operating at the network or application layer, such services often require access to the plaintext of a data stream to perform their processing effectively. The use of end-to-end cryptographic protocols greatly complicates the operation of many active network components. Santos, Spring and Wetherall's packet-caching schemes, for example, are useless if those packets contain encrypted and thus uncacheable data [Santos98, Spring00].

Many active network designs completely ignore the question of data security. Others use a throwaway paragraph to explain how cryptographic protection can be removed or sidestepped to allow data processing, often by severely limiting the location of services or requiring security protocols to be made less secure:

- "in our model, the transcoder is an application level agent that is deployed by the user, within that user's administrative domain. Thus, the user will be able to configure the transcoder with the session key(s) in a secure fashion." [Amir95]
- "end-to-end encryption that covers the entire payload, such as IPSEC, precludes many useful network-embedded services that are already deployed, such as firewalls, transparent proxies, and wireless boosters. For this reason, variant encryption standards that expose header fields are likely to emerge." [Wetherall99]

This thesis examines whether such steps are advisable or necessary. We take a wide look at the field, surveying unconventional threats and investigating active network schemes in three areas: multimedia communication between heterogeneous receivers; efficient TCP communication over lossy wireless links; and the scalable fingerprinting of media streams with the identity of their recipients.

We begin in chapter two by examining the different security protocols in use from the lowest physical layer to the highest human-computer layer in today's Internet. We identify significant overlap in functionality between the network, transport and application layers, and suggest that a combination of network-layer protocols with application-layer assistance provides the most flexible solution for a wide range of application requirements in active services. In chapter three, we show how this combination can be used by multimedia conferencing systems, and describe the protocols used to provide secure conferencing facilities. We also detail the enhancements we have made to these protocols. This gives the background of the area that the majority of active services focus on, and the basis for the services we consider in the rest of the thesis.

Chapter four describes unconventional threats to active services. It considers a number of attacks on data confidentiality and integrity by government agencies and litigation not normally considered in the security literature. Misuse of the judicial discovery process, Customs powers, newly emerging key disclosure warrants and misdirected signals intelligence product are all threats to the confidentiality of the best-protected system. And introducing "reasonable doubt" into the minds of jury members considering evidence of the integrity of data from a "secure" system is much easier than many engineers realise. We show that these attacks place almost impossible security requirements upon active network nodes with access to the plaintext of the data they are processing.

This does not mean that active services will by definition always be insecure. The remaining chapters of the thesis consider three separate problems that can be addressed using active networks. Chapters five and six consider media gateways and TCP performance enhancers, and show that it is possible to achieve similar results using active services that do not require access to the plaintext of data. Chapter seven describes the problems of fingerprinting media distributed to large numbers of recipients, and an active network "watercasting" solution we have developed that can operate on ciphertext streams.

Media gateways have been suggested as a mechanism that allows real-time traffic such as audio and video to be efficiently distributed to a large heterogeneous set of receivers. High-bandwidth flows can be filtered or transcoded to fit over low-bandwidth links in a multicast tree. We show that such filtering can be done on appropriately coded encrypted streams in a

scalable and distributed way that has minimal impact on end users' perception of quality. We further demonstrate that our system can be used to defend against denial of service attacks throughout a network.

TCP performance enhancers have been used to reduce the impact of wireless links on TCP flows. TCP reacts badly to lossy wireless links because it assumes losses are due to congestion and so scales back throughput. Active network solutions reduce loss levels by caching and retransmitting lost packets from base stations and modifying acknowledgements returning to the sender. This cannot work with network-layer security protocols where the acknowledgements are encrypted or authenticated if the base station does not have access to the cryptographic keys used. We instead propose building on link-layer reliability mechanisms whilst ensuring they do not conflict with TCP's congestion control, by providing information on available cache size from the base station to the client. We also show how this can enhance the performance of reliable multicast schemes.

Finally we detail the problems that have been seen with attempts to protect intellectual property using trusted hardware clients, and describe an alternative we have developed: a scalable active service method of fingerprinting data as it is distributed with information identifying the recipient. This allows pirated content to be traced back to the legitimate purchaser that illegally allowed it to be redistributed, and does not rely on a client to act against its owner's interests. It also reduces the amount of trust needed in network nodes, which do not require access to the plaintext of the data.

We conclude by reviewing our work, then taking a wider look at some of the questions it raises. Active networks have been criticised as violating the end-to-end network principles that have been central to the design of the Internet. We believe our work shows that some of the problems used to justify violating these principles can be solved in a way more consistent with the end-to-end philosophy. We consider the future direction of active networks given exponentially increasing bandwidth and connectivity: at least in core networks, this may remove many reasons for in-network services. We also comment on the continuing damage political action is doing to network security, which makes end-to-end security principles as important as ever.

We hope to continue development of the protocols we have created consistent with these principles, and complete this thesis by detailing the further work we have planned.

1.1 Contributions

Our main contributions are as follows:

- Developing a unified model of communications security from the physical to human-computer layer with regard to active services and applying it to secure multimedia conferencing;
- Describing unconventional attacks that real-world active service designers must take account of;
- Showing that a distributed packet filtering service can shape encrypted multimedia traffic to available bandwidth with minimal impact on users compared to media gateways, whilst also providing a defence against denial of service attacks;
- Describing an alternative to TCP error-correcting proxies that gives information to end-systems to allow them to efficiently compose transport and link layer reliability protocols;
- Developing an alternative to trusted hardware clients for intellectual property protection that uses network assistance to provide a scalable distributed fingerprinting service;

- Adding a security perspective to the ongoing debate over end-to-end principles in Internet design.

2 Communications security

*#!/bin/perl -sp0777i<X+d*lMLa^*lN%0]dsXx++lMlN/dsM0<j]dsj $/=unpack('H*',$_);*
$_=`echo 6dio\U$k "SK$/SM$n\EsN0p[lN*1lK[d2%Sa2/d0$^Ixp" |dc`;
s/\W//g;$_=pack('H*',/((..)*)$/)

–Adam Back

2.1 Introduction

Communications security is a field that covers a wide range of techniques for protecting the confidentiality and integrity of transmitted data. These techniques are becoming more important as increasing amounts of sensitive information such as credit card numbers or medical records moves to travel over open networks such as the Internet.

In this chapter, we present an overview of the most important communications security techniques and their impact on active services. Cryptography forms the basis of the majority of these schemes, so is briefly considered before the most important systems and protocols being developed to protect systems in the non-military world are described. It is worth noting that defence agencies the world over are increasingly using Commercial Off-The Shelf (COTS) systems for cost reasons.

Whilst each security protocol has a different focus, their functionality overlaps to a significant degree. Ultimately, all provide the basic security services of confidentiality and integrity. (Ironically, although the Internet is often criticised for being insecure, TCP/IP provides the other basic service – availability – very well). Some provide them at a more general level, usually lower down the protocol stack. Others are very application specific.

Previous work has examined these schemes in isolation, without considering whether protocols could be used together or are indeed redundant given the existence of potentially more suitable alternatives. In this chapter we take an integrated look at protecting information from its source to destination, from the lowest physical-layer protection to the highest human-computer interface designs. Throughout, we consider the benefits and costs of end-to-end security and the impact of different protocols upon active network components.

2.2 Cryptographic concepts

Cryptography has a very long history in military and diplomatic activity [Kahn67]. It allows messages to be encrypted so they are only readable by the intended recipient. It can also be used to create digital signatures for messages, so that the recipient can be confident of the identity of the sender and that the contents of the message have not been altered in transit.

Many historical ciphers, such as that used by Julius Caesar, relied upon the secrecy of their algorithm to protect information. The Caesar cipher simply rotated each character in a message three letters forward in the alphabet to encrypt data, and three letters back to decrypt it. But this security by obscurity fails catastrophically if an attacker discovers the algorithm: every message protected by it becomes vulnerable. As any decryption algorithm used in a modern computer system must be present in every system where data must be decrypted, it is almost impossible to prevent an adversary acquiring such a system and then reverse-engineering the algorithm from its software or hardware.

All modern ciphers rely instead on the secrecy of a small piece of data known as a *key* to protect information. Some *public-key* ciphers use a pair of keys; one to encrypt, and the other to decrypt, data. Both types are also used with hash functions and Message Authentication Codes to protect the integrity of transmitted information, using a secret key to digitally ‘sign’ a message.

2.2.1 Secret-key cryptography

Secret-key or symmetric ciphers use the same key to encrypt and decrypt data. Block ciphers encipher one block of data, usually 64 or 128 bits long, at a time. Stream ciphers produce a keystream that is `exclusive-or`'d with plaintext data to encrypt, and with ciphertext to decrypt.

The most widely-used symmetric cipher is the 25-year old US DES (Data Encryption Standard), despite weak security resulting from its small keysize of 56 bits [NIST88]. Until January 2000, US export controls prevented any product containing a stronger cipher from having an international market, and the resulting reduction in economies of scale discouraged the development of even US versions of such software and hardware. But some US companies, and most international firms, use either Triple DES (applying DES three times with different keys to the same piece of data) or other ciphers with larger keys such as IDEA (International Data Encryption Algorithm) or RC5 [Schneier96]. Triple DES, while slow, is still popular due to its many years of analysis and patent-free status. The US National Institute of Standards and Technology has recently chosen an Advanced Encryption Standard from several competing new ciphers to succeed the DES [NIST01].

2.2.2 Public-key cryptography

Key management – securely distributing the symmetric keys needed by parties communicating using secret-key ciphers – is a difficult problem. Public-key ciphers ameliorate this difficulty, allowing two parties to securely negotiate a key over an insecure link or for each party to publish a public key that encrypts data that can then only be decrypted using a private key kept secure by each party. A public key can also be used to verify a signature made using its related private key.

The Diffie-Hellman key-exchange protocol was developed by the inventors of public-key cryptography, Whit Diffie and Martin Hellman. It allows two parties communicating over an insecure link to negotiate a secret key that even an eavesdropper with full passive access to the link could not compute. Its security is based on the difficulty of calculating discrete logarithms [Diffie76]. It is used in almost every online communication system. The Station-To-Station variant protocol allows the two communicating parties to authenticate each other while negotiating a key by signing various parts of the exchange [Diffie92]. This defends against an active attacker who can alter the data flowing between the two parties during the key negotiation phase.

RSA, named after its inventors (Ron Rivest, Adi Shamir and Len Adleman), is a popular off-line system whose security relies on the difficulty of factoring the product of two large prime numbers [Rivest78]. Its signing function is simply encryption using the sender's private key, verifiable by decrypting using their public key. The other widespread off-line system is the Digital Signature Standard, designed by the US National Security Agency for creating digital signatures [NIST91]. Its patent-free status means it has been used along with another public-key encryption algorithm like Elgamal [Elgamal84] in preference to RSA in recent Internet standards.

Public keys are usually made available as part of a public-key certificate. This contains additional information such as the identity of the key's owner and the uses to which the certificate should be put, and a signature to authenticate this information.

2.2.3 Hash functions

A hash function takes a variable length input and produces a fixed-length output or hash value. Good hash functions are fast to compute, one-way (given an output, it is impossible to reproduce the input) and collision-free (it is difficult to find two inputs that produce the same output – so the output must be critically dependent on every bit of the input).

The two most common hash functions used in cryptography are Message Digest 5 (MD5) [Rivest92] and the Secure Hash Algorithm (SHA) [NIST91]. MD5 is the most recent of a series designed by Ron Rivest, and is widely used in cryptographic protocols. SHA was designed by NSA as part of their Digital Signature Standard, and has recently become more popular due to its perceived greater security (mainly caused by a longer output value and Hans Dobbertin's partially successful attack on MD5 [Dobbertin96]).

Because public-key cryptography is very computationally expensive, the hash of a set of data is usually signed rather than the data itself.

2.2.4 Message Authentication Codes

A MAC is a hash combined with a secret key that is required to compute and verify that MAC. Where two parties share such a secret key, it is a fast alternative to signing data using a public-key cipher. Any hash function can be turned into a MAC by encrypting the hash output with a symmetric cipher, using a provably secure transform such as the HMAC [Krawczyk97].

2.3 Security at each network layer

The Open Systems Interconnection (OSI) model was developed by the International Standards Organisation to enable compatible devices to communicate using standardised protocols [Day83]. It separated communications functionality into seven layers, each performing a distinct function with minimal information required to flow between layers. The first two layers (physical and data link) are concerned with decisions such as the voltages or signals used to represent bits and the data frames that transmitted data is broken up into. The network, transport, session, presentation and application layers are concerned with different software services provided to applications by the network: routing and reliability at one end, application-specific functionality at the other. While OSI can provide a useful conceptual model, few real-world networks follow it closely due to its complexity.

The TCP/IP reference model was developed for the US Department of Defense to allow multiple networks to interconnect and to perform reliably even when large portions of a network was damaged [Cerf74]. It assumes the presence of the physical and data link layers, then adds its own internet, transport and application layers. These map very closely to the OSI network, transport and application layers. The TCP/IP application layer subsumes the small amount of functionality present in the OSI session and presentation layers.

Human-computer
Application
Session
Transport
Network
Data link
Physical

Figure 2.1: model layers

We use a model composed of the OSI physical, data link and session layers and the OSI and TCP/IP network, transport and application layers. The OSI's presentation layer has been omitted: it provided little functionality, and no security schemes have been proposed at this level – ironically, as only political considerations prevented the OSI designers from assigning security to this layer [Tanenbaum98].

Our model also contains a final human-computer layer. If a human is at either end of a communications link, it is important that data is securely transmitted to them from the last communications device, and from their "output" – voice, facial expression, etc. – back into the network.

2.3.1 Physical layer

The physical layer transports raw bits between two points. If access to the medium used to carry bits is prevented or detectable, all data travelling over that medium can be secured.

Almost all users of networks rely on physical security to protect their data, but faith in such protection is rather misplaced. Not only are such networks' cables rarely protected from physical wiretapping, computers on shared access links can use packet sniffing software to perform sophisticated analyses and searches of traffic passing by. While protocols that send plaintext passwords unprotected over such links are still widespread, this creates a gaping vulnerability.

Physically secured lines were widely used before cryptography, particularly public-key, became widespread. The NSA used shielded cables at their Maryland headquarters to protect secure telephone lines [Diffie99]. The British armed forces used cables pressurised with nitrogen and monitored at both ends to detect any deviation from known profiles that might indicate the wires had been drilled into and tapped [Hartley99].

Quantum cryptography can provide a physically secure communication channel that is impossible to eavesdrop on without alerting the communicating parties. It takes advantage of Heisenberg's uncertainty principle, one consequence of which is that you cannot measure the state of a quantum system without affecting it.

If information was sent over a quantum link, an eavesdropper could still access it, even though this would alert the two end points as the eavesdropper affected the link state. Therefore, classical cryptographic techniques are also used. Key material is sent over the quantum line and used in encrypting the data being sent.

Several experimental quantum lines have been built, using optical fibre over several kilometres [Brassard98]. A recent experiment demonstrated a quantum laser link in the open air and suggested that satellites could be supplied with new encryption keys using this technique as they orbited over a base station [Buttler98]. And advances in materials science now allow single photons to be reliably generated, which is important to the security and efficiency of quantum links [Benjamin00].

One potential application is battlefield communications or monitoring, where a direct laser link between participants carries audio and video data over an ad-hoc network. This would also provide better traffic analysis protection than radio communication – extremely important for maintaining surprise and preventing targetting of a signal's origin. But even this would likely be augmented by higher layer security.

Physical security is most useful between two systems with a direct link. While a line may be secure, if intermediate nodes are required to carry the link further, these nodes become targets. But as they have access to unprotected data, they are able to run active services that require access to that data.

2.3.2 Link layer

The link layer turns a raw physical connection into a line free of undetected transmission errors by splitting data into frames, sequentially transmitting them and processing acknowledgements. It works on a hop-by-hop basis, where systems only communicate with their direct neighbours.

Bluetooth is a link-layer radio protocol being designed by Ericsson, Intel and others to provide low-cost short-range radio links. Its major aim is to reduce the need for cables to connect devices such as keyboards or printers to computers, or link a phone and its base station.

Bluetooth provides confidentiality and authentication services at the link layer [Bluetooth99]. Connections can be authenticated in either or both directions using a challenge-response protocol. The verifier sends a random number to the claimant, who replies with a response based on its address, the random number and a secret key shared between the two hosts. Key management is performed at a higher layer, but the current specification calls for short authentication keys to be based on a PIN transferred out of band using a numeric keypad on devices. Privacy is provided using a proprietary cipher with keys between 8 and 128 bits.

The Point-to-Point Protocol is designed for simple links, usually switched circuits or dial-up lines, which transport data between two peers. It allows multiplexing of different network-layer protocols simultaneously over the same link. A Link Control Protocol is used to agree on various options such as encapsulation formats, and to authenticate the peers to each other. Network Control Protocols are used to set up options for encapsulated network-layer protocols, such as assigning an IP address [Simpson94].

PPP authentication is performed using the Challenge Handshake Authentication Protocol [Simpson96]. The host being authenticated is sent a random challenge value, which it combines with a session identifier and secret known to both peers then returns. The challenging host also calculates this value and checks that it matches the value returned. CHAP can be re-run at any time during the connection. If the returned authentication value is incorrect, the connection is terminated.

The Encryption Control Protocol allows PPP's data frames to be encrypted using a negotiated algorithm [Meyer96]. Once the basic link options have been configured, either of the peers may send a `configure-request` frame to the other containing a set of algorithms they are willing to use to decrypt data. The receiver will reply with the algorithm it is willing to use, or close the link if none of them are acceptable. From that point on, the data within each frame from the receiver to the sender is encrypted with the agreed algorithm. The same process is carried out to determine the algorithm used in the other direction, which is not necessarily the same as the first. At any point when one of the peers cannot decrypt received data, it sends a `reset-request` message then discards any further received encrypted frames. The recipient of the request resets itself to an initial state and sends a `request-ack`, which resets the state at the other end of the connection. The peers then begin communicating securely again.

Link layer security is very difficult to use successfully outside a tightly controlled organisation such as the armed forces. It works only on a direct link, requiring strong protection for intermediate nodes in a network. It is marginally useful for confidentiality between peripherals such as a Walkman and its headphones, but for other applications end-to-end security at a higher layer is far more useful. Its main use is authenticating devices to each other; a cordless phone to its base station, for example. And as with the physical layer, intermediate nodes are able to provide active services with plaintext access.

Physical and link layer security's biggest advantage is that they transparently protect streams without requiring software producing those streams to be modified. This is particularly useful for low-capability devices such as headphones that would be overloaded by processing-intensive security code. Key management is also greatly simplified: only the two nodes on the end of a link need share a key, which can be changed independently of any other node on the network. And traffic analysis is difficult, as all routing and structural information in the traffic is hidden.

2.3.3 Network layer

In packet-switched networks such as the Internet, data are grouped into packets that are independently routed to their destination. If the network drivers providing this service can provide security, data streams will be transparently secured with no need to modify applications.

The Internet Protocol Security (IPSEC) working group of the Internet Engineering Task Force is completing standardisation of security mechanisms for IP packets. Their work is a mandatory part of IPv6 and can also be supported using option fields in IPv4. The specifications provide an algorithm-independent framework into which specific cryptographic methods can be inserted [Thayer98].

Two mechanisms are used to protect packet data. The Authentication Header (AH) allows data to be signed, assuring its authenticity and integrity but not secrecy, as shown in figure 2.2

[Kent98a]. The Encapsulating Security Payload (ESP) provides confidentiality, as shown in figure 2.3 [Kent98b]. Both mechanisms can be used in tunnel mode (an entire IP packet is encapsulated within another before applying the security services) or transport mode (only higher-layer data is secured). ESP mode can also incorporate authentication procedures, but an authentication-only mode was politically convenient: US export restrictions on cryptography only apply to systems that provide confidentiality. AH also allows the protection of immutable and predictable IP headers, which ESP cannot provide in transport mode.

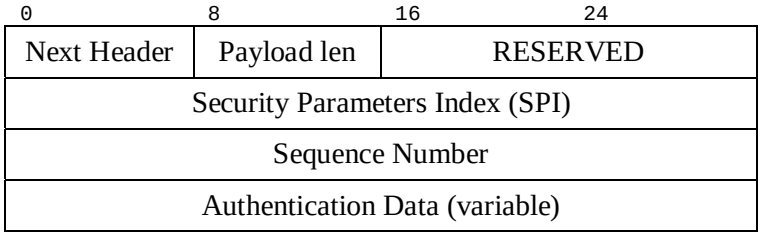


Figure 2.2: Authentication Header format

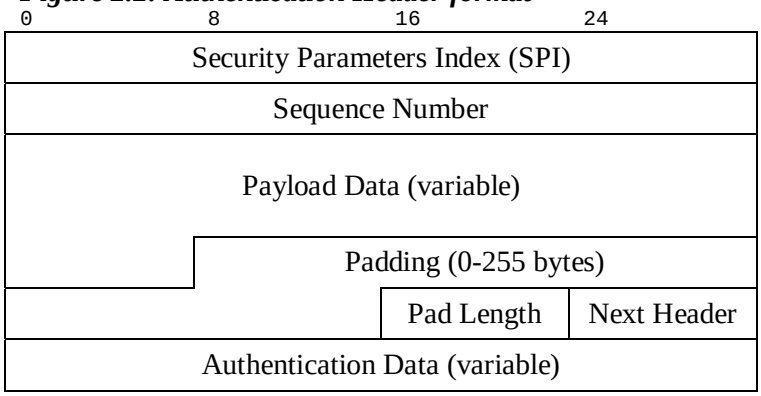


Figure 2.3: Encapsulating Security Payload packet format

The Authentication Header is placed between the IP and higher layer headers of a packet, and contains information that the recipient can use to authenticate the sender and contents of the rest of the packet. This is more efficient than applying transformations to the entire packet. An anti-replay service can prevent old traffic being resent to a host, using the sequence numbers in the authentication header.

Data can be placed in an Encapsulating Security Payload before it leaves the sender, hiding its contents until it reaches the receiver, who decrypts it using the secret key they share with the sender. In tunnel mode, the source and destination of the encapsulated packet can be hidden, so preventing some traffic analysis attacks. The padding used to ensure the data being encrypted is the correct length for use with a specific cipher can also be extended to conceal the true length of that data, providing further traffic flow confidentiality. Transport mode is simpler, and normally used between end systems. If a security gateway is at either end of a connection, tunnel mode must be used. Tunnel mode is also less disruptive to the current Internet infrastructure, as packets look entirely normal to routers. Only at the end of their journey are the encrypted and authenticated contents of a packet decapsulated. Security gateways can forward such packets to hosts without IPSEC support.

AH and ESP both require communicating hosts to share secret keys to authenticate and encrypt transmitted data. It is relatively simple to manually configure hosts with fixed keys, but this is completely unscalable. Hosts need also to agree on the cryptographic systems they both understand.

The Internet Key Exchange (IKE) allows two hosts to agree on these parameters [Harkins98]. After setting up a secure ISAKMP (Internet Security Association and Key Management Protocol) link [Maughan98], IKE hosts generate keys for, and negotiate, IPSEC Security Associations. Each association is used by network code to select the transformations it will apply to each packet. The exchange is finally authenticated to prevent a man-in-the-middle

attack, and optionally identify each host. Various types of public-key certificate can be used at this stage to increase the security of the authentication.

AH and ESP are relatively easily extended to multicast packets, but key management is much harder for multicast groups. While protocols exist to allow group key negotiation or distribution [Hardjano98a], they are not scalable and thus defeat the purpose of multicast – particularly its ability to rapidly change group membership. They also reduce security by providing central points of attack.

One suggested solution divides a multicast network into trunk and leaf regions. Each leaf region contains one or more key management entities, which manage the security of hosts within that region. The trunk region consists of the "border" key manager from each leaf region. As traffic passes from a host through its leaf region, the trunk region and then into the other leaf regions, an active service key translator decrypts and re-encrypts traffic appropriately [Hardjano98b].

Region separation increases the scalability of the scheme, and allows different regions to use different intra-group key management protocols. Within both regions, members can be added and deleted at will, and keys changed, without affecting hosts in other regions. Key managers must run on trusted hosts (specified by the traffic originator in a one-to-many flow, or democratically agreed among leaf members in a many-to-many situation) but can therefore use long-lived keys, while allowing leaves to rapidly and independently give new keys to their members. This point could give plaintext access to other active services.

The main advantage of security at the network layer is its transparency to applications. If two communicating hosts are running IPSEC-enhanced drivers, any links between them can automatically be secured. As IPSEC is a mandatory part of IPv6, this will eventually become the norm. Connectionless sessions such as videoconferencing using the User Datagram Protocol can also be protected much more easily at this layer than at the transport layer. Traffic analysis protection can be automatically applied to all data flows rather than requiring separate applications to provide their own schemes.

By default, different processes owned by multiple users on the same machine will use the same encryption keys and access policies for links to the same machines. This host-based security is ideal for implementing virtual private networks, which many firewall systems are now starting to do [Oppliger97]. But, subject to approval by the administrator of a machine, IPSEC-aware services and applications at higher layers can also use enhanced interfaces to the network layer to allow individual control over the operation of IPSEC, with user-oriented keys, algorithm selection and other option choices. This gives a flexibility more usually associated with the transport layer.

While the network layer is not traditionally end-to-end, IPSEC's encapsulation and tunnelling modes can provide end-to-end security by providing the information required to route packets through the network in cleartext packet headers. This complicates the operation of active network components such as firewalls or Network Address Translators, which require access to information inside the encrypted packet data. The simplest solution to this problem is for security to switch from the network layer at the firewall or NAT to the link or physical layers inside the affected network. This also helps organisations that require cleartext access to data travelling into or out of their networks.

Traffic analysis protection is more difficult with end-to-end schemes, but can be supported using active services that do not require plaintext access (and indeed use multiple layers of encryption). Onion routing, for example, transports information through a series of anonymising nodes that hide traffic information as well as the information itself [Reed98b]. Clients set up routes through a number of nodes (usually five) and encrypt data using different keys for each node. As it passes through each router, a layer of encryption is "peeled off" before the data is forwarded to the next router in the chain, mixed and re-ordered with other traffic travelling between the two routers. Nodes can also send each other random padding

data to prevent attackers using periods of inactivity to attempt to trace data passing between different routers. This makes it very difficult for an attacker to track traffic passing through the network, even if they have compromised a number of routers and can monitor all data flows.

2.3.4 Transport layer

The transport layer is responsible for splitting information from the session layer into smaller units and passing it to the network layer for transport to the other end of a connection. If high throughput is required, data might be multiplexed across several connections. If creating or maintaining a network connection is expensive, several connections might be multiplexed into one. The transport layer is the first true end-to-end layer, where a program on one machine talks to a similar program on another, rather than a host talking to another host or router. A naming service is also provided so a process can describe with whom it wants to communicate.

The Secure SHell (SSH) architecture was designed as a secure replacement for the session-layer `telnet`, but is based on a secure transport layer protocol [Ylonen99]. Data is sent over a reliable network connection in blocks of up 35000 bytes (or larger when both parties agree), and can be compressed, encrypted, and authenticated with a Message Authentication Code. The keys used to encrypt and authenticate the data packets can be negotiated when the link is created, and authenticated either way using one of several public-key algorithms. After every hour or gigabyte of data sent, the two parties choose a new key using the same algorithms. Random cover data can be sent at any time in packets marked "ignore" to protect against traffic analysis.

Secure Sockets Layer is probably the most widely used security standard on the Internet. It is a transport-level protocol designed by Netscape to enable secure communication between a Web browser and server. Almost all secure Web access takes place over a SSL connection.

TLS (Transport Layer Security) is the standardised successor to SSL [Dierks99]. Its goal is to provide privacy and data integrity between two communicating applications. Its two main protocols are the record protocol, which provides a private and reliable connection, and the handshake protocol, used to authenticate client and server and negotiate cryptographic algorithms and keys.

The record protocol fragments data into blocks of 2^{14} bytes or less. Each block can be compressed, encrypted, and authenticated using a Message Authentication Code. TLS and SSH both allow the use of stream as well as block ciphers even though both are packetizing protocols. This allows a stream of data to be encrypted or decrypted as each byte arrives for processing, reducing latency – particularly important for interactive applications. A key calculation algorithm is used to generate keys, Initialisation Vectors for the encryption and MAC secrets from secret parameters supplied by the handshake protocol.

The handshake protocol negotiates each session: a session identifier, compression method, cipher specification (encryption and MAC algorithms), 48-byte master secret, a resumable flag and optional certificates for either party. A resumable session can be used to set up several connections. A session can be renegotiated at any time during a connection. Alert messages of varying levels may be sent; "fatal" alerts cause the connection to be terminated and session invalidated for future connections.

The Domain Name System is the Internet's naming mechanism, mapping addresses such as `ietf.org` to a numeric IP address. Security extensions have been developed to allow authentication of the information supplied through the DNS [Eastlake99]. Secure DNS servers send signature records along with information requested by clients, allowing clients configured with the relevant public keys to verify server replies. Clients can also securely request public keys of servers as long as there is a trusted path between the client and server, or the server has a certificate acceptable to the client. Certificates for other protocols such as

e-mail, Web access or IPSEC can also be supplied using this mechanism. The operation of the DNS protocol itself can be authenticated, and with an auxiliary mechanism [Eastlake97], DNS information can be dynamically and securely updated.

Transport layer security is not as transparent as network layer protection. Inter-process communication libraries must be updated, and applications must be updated if they are to make use of any security beyond the default provided by those libraries. It is much easier, however, for applications to be distributed to users with transport layer security services. Netscape can distribute its secure sockets code with its browser and provide secure connections without requiring users to install an IPSEC-aware network driver. Process-to-process security is also the default mode of operation, rather than the network layer's host-to-host [Oppliger97].

End-to-end security at this layer allows network-layer active services such as TCP performance boosters to function, but prevents services such as 'transparent Web proxies' from accessing the application-layer data they require. Explicit Web proxies generally do not cache secured content because it may contain especially sensitive information such as credit card numbers, and is usually personalised for one receipient. Because the DNS works in a specifically hierarchical manner, with queries directed up a tree toward the authoritative nameserver for a given domain, DNSSEC automatically uses 'hop-by-hop' protection of links between client and nameservers.

2.3.5 Session layer

The session layer enables users on different machines to establish sessions that provide ordinary data transport and enhanced services useful in some applications. It is unclear whether mail transport protocols (SMTP, POP, IMAP etc.) belong here or at the application layer; but it is generally agreed that `telnet` is a session-layer protocol.

Blaze and Bellovin surveyed secure `telnet` implementations and designed their own that used standard `telnet` options [Blaze95]. After discovering that the daemon at the other end of a connection supports security extensions (using the WILL/WONT, DO/DONT mechanism), their secure `telnet` negotiates a key using the SEND/IS protocol. A one-way function of the agreed key is presented as a challenge to the user, who calculates a response with a hand-held authentication device. Traffic between the two hosts is then encrypted.

The two authors also designed an encrypted session manager. `esm` exploits the BSD "pseudo-tty" mechanism to run secure sessions over IP and other connections such as `tip`, `kermit` and `datakit`. A user runs it on their local host before remotely logging in to another system and running `esm` in server mode on that machine. The two processes negotiate a key and then encrypt further traffic. The only authentication provided is a one-way function of the agreed key in the remote environment. By operating at this layer, `esm` can provide security across application-layer firewalls and multi-hop logins.

SSH runs two session-layer protocols over its transport layer protocol: the user authentication and connection protocols. They allow secure remote login and secure port and X11 forwarding over insecure network connections.

The authentication protocol begins with the server sending the client a list of the authentication methods it supports. It may also send a message to be displayed to the user warning that only authorised users may connect to the server, important for later prosecution of intruders in some jurisdictions. The client chooses an authentication method it also understands, and replies with a user name, service name, and authentication data. The method may be public-key based, where the authorisation request is signed with the user's private key, or password-based, where the user's password is sent. A password must not be used if the underlying transport protocol does not provide confidentiality. Servers may also accept connections authenticated by an authorised host's private key, optionally where the user name on that host matches a user name on the server.

The connection protocol allows interactive login sessions, remote execution of commands and forwarded TCP/IP and X11 connections. It can multiplex all links from a client to a server into one encrypted connection to impede traffic analysis. It also allows communication with an authentication agent, which can cache passphrases for a user-specified time to reduce the need for them to be entered every time a new connection is opened.

The session layer is an oddity, really only encompassing remote terminal facilities. As popular operating systems have largely changed over to graphical user interfaces, it has become less important. Security at this level is mostly useful for securing `telnet` and its successors such as SSH, providing transport-layer services with a small number of enhancements to ease the securing of terminal and X11 software. X-Windows is unfortunately unlikely to be widely adopted outside the Unix world without built-in support in Microsoft Windows, but a secure version would be extremely useful in both operating systems.

Because the session layer is generally used to provide individual point-to-point connections, active services at this layer provide little benefit, and so far none have been proposed.

2.3.6 Application layer

The application layer contains a vast number of protocols defined and used by applications to communicate with each other. Web browsing, electronic mail and videoconferencing services are just a few examples. The common thread is that processing of an application-level protocol requires an understanding of the semantics and not just syntax of the data passing through.

Many application protocols are now being extended to operate securely. The Internet Engineering Steering Group insists that all new Internet protocols contain a section on security considerations [Postel97]. Some protocols are designing their own security extensions, while others are providing standardised ways to access lower-layer security.

Secure Web access is an interesting example of both. The Secure Hypertext Transfer Protocol extends HTTP to allow documents to be signed and encrypted [Shostack95]. HTTP over TLS runs a normal HTTP session over a TLS connection on port 443 [Rescorla98]; TLS over HTTP uses HTTP options to initiate a TLS link on a current HTTP connection [Khare98]. SHTTP allows fine-grained control over the security of individual documents, although an application could use a combination of TLS and HTTP to achieve the same effect.

E-mail remains the most widely used communication software on the Internet. S/MIME (Secure Multipurpose Internet Mail Extensions) [Ramsdell99] and PGP (Pretty Good Privacy) [Callas98] are competing to become the standard to authenticate and provide privacy for electronic messages. Both use MIME and similar public and private-key cryptographic algorithms to encrypt and sign parts of messages. S/MIME is more standards-based, using existing X.509 certificates, cipher modes and MIME transport encodings, while PGP uses its own formats. Unfortunately the two are not interoperable. Port 465 has also been assigned to run the Simple Mail Transfer Protocol over TLS between SMTP servers, which protects mail between but not on mail servers. Ideally this would be used to deliver mail directly from the sender's machine to an SMTP server running on the recipient's workstation [Brown00a].

Videoconferencing tools are being gradually extended to allow secure meetings. The Session Announcement and Initiation protocols (SAP and SIP) allow multimedia sessions to be announced and participants invited using two simple ASCII-based protocols. Extensions have been defined to allow authenticated and private announcements and invitations using S/MIME and PGP formats [Handley00, Handley99b]. These messages can also contain keys used to encrypt the conference material itself using the Real-time Transport Protocol [Schulzrinne96]. RTP does not provide authentication services, and originally expected all of its security capabilities to be superseded by services provided by IPSEC once that becomes widespread. A Secure RTP standard is now being defined that allows RTP packets to be

encrypted and authenticated whilst still allowing header compression, which is useful for low-capacity wireless links [Blom01].

Numerous application-layer schemes are also being designed to allow anonymous browsing and messaging by preventing traffic analysis. *Crowds* routes HTTP requests around a number of nodes running its proxy software; each node randomly decides whether to forward a received request to its destination or to another node [Reiter98]. E-mail *mixes* strip identifying headers from messages and route them via several nodes that delay, re-order, and pad outgoing data [Chaum81]. Unlike network-layer traffic protection, application-layer schemes can check the content of traffic for identifying information as well as protect its transmission. Therefore application-layer protection is often applied in conjunction with network-layer schemes.

Before transmitted information has been cryptographically protected at the originating machine, and after it has been decrypted at its destination, it is vulnerable to malicious software. Memory management in operating systems is vital to prevent one application eavesdropping on the data of another. Trusted windows, display drivers and Graphical User Interfaces are all necessary to display secure information. But the protection of computer systems is a topic at least as large as communications security, and will not be discussed further here. Interested readers might start with the US government's standards for secure systems to gain some appreciation of the task involved [DoD85].

If information cannot be adequately protected in end systems, it can at least be traced if compromised using *fingerprinting*. This watermarks each copy of a piece of data with details of its recipient. Almost all watermarking schemes work at the application layer. They manipulate low-significance bits, compression variables or other signal characteristics in audiovisual data, inter-word and line spacing in text, and other datatype-specific parameters. This obviously requires application-specific knowledge in marking algorithms. Unfortunately, simple attacks such as introducing jitter or resampling data defeat almost all current watermarking schemes, and other designs can be broken with slightly more effort [Petitcolas98]. Hopefully, the next generation of algorithms will be more robust.

Watermarking large numbers of data streams is a processing-intensive task, and ideally suited to distribution throughout components in the network being used to distribute that data. We describe a scheme called watercasting in chapter seven that performs this distributed watermarking without requiring those components to be trusted and have access to the plaintext of the data [Brown99a].

Application layer security has an advantage where separate services are required for different pieces of information. Lower layer security can set up secure links between hosts and processes, but can only provide coarse-grained services for information flowing over those links [Oppliger97]. HTTP allows different components of a Web page to be fetched using different connections, so a client could request different security services for different components simply by setting up transport or network layer secure links separately for each one. But SHTTP allows different security properties to be applied to parts of the same file. Servers may wish to sign relatively slowly-updated data with a more secure off-line key, but sign constantly changing parts of the same page with a less secure on-line key, for example.

Being built into applications, this security layer also requires no interaction with services outside an application. This makes it easy to deploy, but difficult to write. It is much easier for an application to request a set of security services from an inter-process communication library or IP stack than to provide and maintain all of its own security code. It is also difficult for non-specialist application designers to design secure protocols, particularly without the scrutiny of the many security engineers who contributed to the development of standards such as IPSEC.

This reduces the reliability and security of application-layer solutions. It can also take time for a set of inter-operating programs such as e-mail clients to be fully upgraded with security

code. Application-layer proxies that provide security services such as encrypting and signing e-mail at the user's workstation without requiring any modification of other software can ease the transition to this state [Brown99b].

End-to-end application-layer security can work with some active services that do not require access to protected application data such as Web caches, but not others that do such as transcoders.

2.3.7 Human-computer layer

Secure communication involving a human requires a secure link between computer and human inputs and outputs. Since humans are not equipped with cryptographic processors (although this would be an interesting application of neural implants!) other mechanisms must be used to ensure the confidentiality and authenticity of transmitted information.

Computer monitors give off a small but significant amount of electromagnetic radiation while displaying images. Signals can also be picked up from power lines. So-called Tempest emanations can be detected even by low-cost TV equipment and used to reconstruct a monitor image at a distance [Eck85]. Hardware Tempest protection requires expensive metallic shielding of equipment or the rooms containing it [US90], and so is rarely used outside the military and diplomatic worlds.

To provide some degree of protection to those with smaller budgets, Kuhn and Anderson developed 'soft Tempest' fonts that make text information hard to detect and reconstruct by removing high-frequency components and putting random variations into characters [Kuhn98]. PGP 6 contains a secure viewer that uses these fonts to display particularly sensitive information. Kuhn suggests that their techniques could be implemented on graphics cards, providing protection for all software running on a machine. It would also be easy to add to display drivers or fonts, again providing protection without requiring individual applications to be modified.

Video and particularly audio output are available to an attacker close by a system. With remote access laptops or PDAs, confidential information could be overheard or seen in a public area such as a train. Simple hardware devices like micro-louvre screens that reduce the angle from which a monitor can be viewed, or headphones, reduce these risks. But stronger protection is available from devices that limit the locations at which information can be viewed. If users can only access sensitive data from restricted areas, it is much less likely that attackers will have the physical access to eavesdrop on their data – either in person or using bugging devices. It can also limit the ability of users to display confidential information to unauthorised recipients. Denning and MacDoran describe a system that uses the Global Positioning Satellite network to provide an unforgeable location signature from a remote device [Denning96]. It uses the unpredictability of GPS signals due to satellite orbit perturbations and the signal instabilities added by the US Department of Defense to prevent spoofing. The location information can be updated every few microseconds if required.

Similar considerations apply to information going from human to computer, with the added concern of authenticity. The strongest authentication software is no use if the information it is authenticating comes from an attacker.

Most current systems require users to log-in to a system using passwords, which are notorious for being badly chosen and used. An interesting variation is passfaces, where users are required to remember and then pick out certain faces during the log-in procedure [Davies94]. This takes advantage of the very good human memory for faces, and the difficulty in accidentally or deliberately disclosing the secret faces. Higher assurance systems use hardware such as smartcards to authenticate users. But all these systems suffer from the problem that once users are logged in they may leave their computer, allowing an attacker with physical access to send information authenticated as coming from that user.

A better solution is to continually authenticate the user with one or more of their inputs. Everyone has a distinctive typing style that can be recognised by software, so keyboard input can be checked [Shepherd95]. Keyboards and mice could even have fingerprint recognition built in. Automatic voice and face recognition systems can validate multimedia input [Cope90] – it has been suggested that cellphones should simply stop working when used by an unauthorised speaker. And iris scans have proven to be very accurate discriminators [Daugman94], and could be made part of facial recognition techniques.

Security of the human-computer interface is absolutely vital. Information can criss-cross the planet safely using links secured at any of the lower layers, but be compromised at the first or last hop into or out of the network. Some applications are starting to use the techniques outlined in this section, but most would be best applied by the operating system in order to benefit all applications without requiring each to be modified. Combining soft Tempest with location-based access control should greatly reduce the threats to confidentiality of information, while continual biometric authentication of users ensures that incorrect information is not inserted into a system before lower layer cryptographic techniques are used to protect its authenticity.

2.4 Conclusion

In this chapter, we have examined security protocols from the lowest physical layer up to the human-computer interface. Because they protect different information at each protocol layer, their effect on active services is varied.

There are many schemes to provide confidentiality and authentication services to transmitted data at each of these layers. The methods of the two lowest levels are becoming decreasingly useful as the complexity of networks increases, particularly since their primary benefit of traffic analysis protection can now be provided at higher layers with active service assistance. There is little to choose between network and transport-layer protection for general data streams. As IPSEC becomes a normal feature of all networked systems and develops standard interfaces for applications, it seems there will be less need for transport security – although it may remain popular due to its prevalence in Web browsers. It can also be used with network layer protocol boosters such as TCP snoop. Session layer services are likely to be replaced by higher-level application protocols running over lower-level secure streams.

Application-layer security will remain important for its ability to apply different services to different parts of data, although it can be replaced in many programs by standardised methods of using lower-level services, as anticipated by RTP [Schulzrinne96]. E-mail security will remain the most important application-layer service due to different requirements for different sections of messages. Its store-and-forward nature also means that it is very difficult to use lower-level security services, as no direct link is made between the sender and recipient of a message. Using short-life encryption keys and applying additional lower-layer security to the links between mail servers can reduce the security vulnerabilities this introduces [Brown00a].

Protection of information once it has reached its destination is a more difficult problem. Requiring trusted hardware or software (the latter an oxymoron on insecure platforms like Windows) at all clients to restrict the redistribution of data seems to be a battle that has already been lost when there are more than a few authorised recipients outside a tightly-restricted organisation like the armed forces. New systems seem to be broken almost as soon as they are released, and the economic incentive for breaking such protection can only increase as the value of information being transmitted across networks grows.

Fingerprinting seems the best current solution. While it cannot prevent information being redistributed, it does provide an audit trail that allows the redistributor to be caught and punished via the legal process. Once schemes are available that survive simple attacks [Petitcolas98] they can be applied at the application-layer by the source of data, possibly operating in conjunction with trusted network components at the network layer to distribute the processing and distribution load as described in chapter seven [Brown99a].

Other current active network services will be difficult to use once the data streams they are processing use end-to-end encryption. If in-network components were provided with the plaintext of data, either through using physical or link-layer protection or some mechanism to make them one of the end points using end-to-end encryption, they would become valuable targets to attackers. Services must adapt to allow the processing of encrypted streams. This would be easiest using IPSEC, possibly with extra non-security-critical information in cleartext packet headers.

Firewalls are caused great difficulty by encrypted data streams. There is no way they can check information flowing through them if its plaintext is not available. Organisations may consider it vital to stop viruses or hackers entering their networks, or sensitive information flowing out. This is best achieved by configuring clients to use one IPSEC link to the organisation's firewall, which constructs another to the destination outside the network. Physical-layer protection of the internal network also reduces the risks to information it carries. This is one example of data moving between trust domains: different threats require different responses. More sophisticated schemes label information according to its sensitivity, and will not allow highly-sensitive data to travel over networks without adequate protection [NIST94].

Each layer hides different traffic information. SSH can multiplex connections, hiding the port numbers of UDP and TCP connections. IPSEC can hide all higher-level protocol information in tunnelling mode. Powerful traffic analysis protection has been an important advantage of physical and link-layer security schemes, but the work of Chaum [Chaum81] and others has allowed active services to provide this service to higher-layer schemes.

E-mail and Web application-layer traffic protection is now in widespread use. Network-layer schemes are starting to become more popular. They provide protection to all higher-level services, but cannot check the content of transmitted data for identity information. This is important if data should be anonymous to the recipient as well as attackers *en route*. A combination of general network-layer protection with specific application-layer checking of mail and News messages, Web form data, etc. at source therefore seems best, and is used by the Onion Routing system [Reed98b].

Considering all these factors, it is obvious that there is no "best" layer at which to apply security. Designers must decide which factors are most important to their system. Higher, end-to-end protection is better for security but requires extra mechanisms to provide the traffic analysis resistance present in low-layer schemes. Different services require different combinations of stream and data security. It seems however that a combination of network and application-layer security can meet most distribution requirements, and also support data traceability and traffic analysis protection using active services without plaintext access. We show in the next chapter how this combination can be used in secure conferencing systems.

3 Secure multimedia conferencing

"I want to say a special welcome to everyone that's climbed into the Internet tonight, and has got into the MBone — and I hope it doesn't all collapse!"

— Mick Jagger

3.1 Introduction

Because of the high bandwidth requirements of multimedia data, much active network research has concentrated on allowing heterogeneous clients with very different resources to conference together with optimal results. In this chapter we therefore describe the architecture used for lightweight multimedia conferencing over the Internet as a background to later chapters, and the developments we have made to increase its security. Chapter five describes current active network conferencing systems and our secure alternative.

We first examine the basic mechanisms by which audio, video and shared whiteboard data is transmitted between the participants in a lightweight IP conference. This includes the tools and data formats that are used. We then consider how sessions are set up, modified and torn down. Finally, we consider existing mechanisms for securing conference content and signalling and suggest improvements based on our analysis of security protocols in chapter two.

3.2 Basic conferencing architecture

The two defining problems of conferencing are communication between (possibly large) groups of people and real-time delivery of information. In the Internet, these capabilities are supported at a number of levels.

Several different tools can be used to send and receive conference data. *vic*, shown in figure 3.1, supports video. *rat* and *wb* process audio and shared whiteboard data. Support for multicast conferencing is also now beginning to appear in commercial products such as *RealPlayer* from Real Networks.

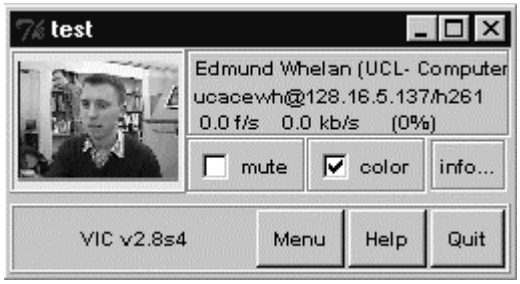


Figure 3.1: the VIdeoConferencing tool

The people participating in a conference need some idea of the context in which the conference is happening, which can be formalised as a conference policy. Some conferences are essentially crowds gathered around an attraction, while others have very formal guidelines on who may take part (listen in) and who may speak at which point. Initially the participants must establish communication relationships (conference *set-up*). During the conference, some conference control information is exchanged to implement a conference policy or at least to inform the participants who is present.

3.3 Transport mechanism

Conference data must be distributed to all participants. Early systems used a fan-out of data streams, setting up a connection between each pair of participants, which meant that the same information crossed some networks more than once. The Internet architecture uses the more efficient approach of multicasting the information to all participants.

IP multicast provides efficient many-to-many data distribution in an Internet environment [Deering89]. Multicast is a natural solution for multi-party conferencing because of the efficiency of its data distribution trees, with data being replicated in the network at appropriate points rather than in end-systems. It also avoids the need to configure special-purpose servers to support the session; such servers require support, cause traffic concentration and can be a bottleneck. Receivers do not need to know who or where the senders are to receive traffic from them. Senders never need to know who the receivers are. Neither senders nor receivers need to care about the network topology as the network optimises delivery. The portion of the Internet that supports multicast is often referred to as the Mbone.

Multimedia conferences require real-time delivery of streamed continuous media (audio and video data) and reliable, near real-time delivery of shared workspace information. In a datagram network, multimedia information must be transmitted in packets, some of which may be delayed more than others. To allow audio and video streams to be played out at the recipient with the correct timing, information must be transmitted that allows the recipient to reconstruct the timing.

The Real-time Transport Protocol (RTP) provides all these services [Schulzrinne96]. It provides a standard format packet header that gives a media specific timestamp, as well as payload format information and sequence numbering amongst other things. RTP is normally carried using the User Datagram Protocol. It does not provide nor require any connection set-up, nor does it provide any enhanced reliability over UDP.

Continuous-media tools like *rat* and *vic* do not normally provide error re-transmission facilities; they can reconstruct missing audio and video at the receiver, sometimes helped by Forward Error Correction (FEC) data to correct errors incurred during transmission. By contrast, shared-workspace tools like *wb* (whiteboard) require fully reliable transmission. To achieve this there is a need for a reliable multicast transport. Various techniques to achieve this have been implemented. For RTP to provide a useful media flow, there must be sufficient capacity in the relevant traffic class to accommodate the traffic. How this capacity is ensured is independent of RTP.

Each original RTP source is identified by a source identifier, which is carried in every packet. RTP allows flows from several sources to be mixed in gateways to provide a single resulting flow. When this happens, each mixed packet contains the source IDs of all the contributing sources. RTP media timestamps are in units that are appropriate to the media flow. For example, 8KHz sampled Pulse Code Modulation-encoded audio has a timestamp clock rate of 8KHz. This means that inter-flow synchronisation is not possible from the RTP timestamps alone.

Each RTP flow is supplemented by Real-Time Control Protocol (RTCP) packets. There are a number of different RTCP packet types. RTCP packets describe the relationship between the real-time clock at a sender and the RTP media timestamps so that inter-flow synchronisation can be performed, and they provide textual information to identify a sender in a conference from the source ID.

The penetration of multicast packets can be limited by two mechanisms: *administrative scope* and *time-to-live*. The first ensures that packets for a specific multicast group are constrained to stay inside a set of routers that have been configured to limit the flow. The second ensures that the packets are constrained by the total number of routers the packet can traverse before being discarded

It is important to include mechanisms for co-ordinating and controlling different streams of data. The Real-Time Streaming Protocol (RTSP) [Schulzrinne98] has been defined for controlling activities between systems, while the Message Bus (Mbus) controls activities inside single systems [Perkins99b].

3.4 Packet multimedia formats

Various data formats are defined to carry different media types using RTP. Each is described in an RTP *profile* document.

3.4.1 Audio

The primary audio format used to carry voice by many Internet conferencing applications, including *rat*, is taken from the GSM mobile telephony standard. Its Regular Pulse Excited – Linear Predictive Coder (RPE—LPC) encodes 20ms samples as a 260-bit combination of previous samples with a residual, producing a 13kbps stream [Redl95]. This combines reasonable speech quality with low bandwidth and complexity, allowing resource-limited implementations.

RTP allows redundant lower-quality secondary encodings of samples to be carried in packets following the original sample. This allows the receiver to improve the perceived sound quality in the presence of packet loss by filling in gaps with the secondary samples [Perkins97]. The receiver can also conceal packet loss by interpolating from surrounding packets [Perkins98].

3.4.2 Video

A number of standards exist across for transmission of video data across a network. The most important are MPEG and H.261.

The Moving Pictures Experts Group (MPEG) was set up by the International Standards Organisation to develop a video and associated audio compression scheme at around 1.5Mbit/s. Video is coded as one of three frame types: Intrapictures (I), Predicted pictures (P) and Bidirectional pictures (B). I-frames provide moderately-compressed reference frames. P-frames contain motion vectors based on a previous reference frame along with a residual. B-frames are coded in the same way with a previous and future reference frame. All three types use lossless compression to further reduce the resulting data.

A video stream is made up of a sequence of I-, P- and B-frames. The ratio between each type depends on application requirements: random access and editing require more I-frames. But typically, B-frames will constitute 75% or more of a stream due to their high compression level [LeGall91].

H.261 was designed by CCITT to carry video over p*64kb/s ISDN lines (where p is between 1 and 30). Higher values of p (≥ 6) support good-quality video in the Common Intermediate Format (CIF). This allows up to 29.97 frames per second at a resolution of 288*360 pixels. Lower bandwidth links use Quarter-CIF at 144*180 pixels. Encoders may drop 1, 2 or 3 frames between each transmitted frame.

At this resolution, CIF and QCIF video has a bandwidth of 36.45 and 9.115 Mbit/s. To reduce this to the kilobit range, a series of steps are taken. Each frame is divided into a hierarchical structure of Pictures, Groups of Blocks (GOBs), Macro Blocks and Blocks. A lossy coding scheme removes spatial and temporal redundancy from Blocks and Macro Blocks using DCT and Differential Pulse Code Modulation. A variable word-length entropy coder is then applied in a final lossless compression phase [Liou91].

H.261 is extremely vulnerable to packet loss because it assumes an underlying bit stream with isolated and independent losses. Few error recovery procedures are provided. Lost data requires a receiver to resynchronise its compression parameters, block addresses and prediction state. The start of the next GOB resynchronises the entropy and address codes, but a potentially large amount of data up to that point must be discarded. To reduce bandwidth requirements, the sender rarely supplies the intra-coded blocks needed for temporal resynchronisation – sometimes as little as once every 132 frames.

Intra-H.261 is a variant designed by Steve McCanne to be resistant to the burst erasures common on the Internet. These introduce two problems: a lost packet will prevent the decoding of any other macroblock contained in that GOB; and losses will affect every further macroblock received until the next resynchronisation macroblock is received. Creating burst loss-resilient codes unacceptably increases latency for videoconferencing due to the number of packets data needs to be spread over.

Therefore, Intra-H.261 uses conditional replenishment rather than differential updates of macroblocks. Image blocks are compacted using a Discrete Cosine Transform, quantised to reduce the entropy of quantised coefficients, and then Huffman compressed. Only blocks that change by more than a threshold value are updated, using a full intra-coded block rather than temporal prediction. GOBs are not used: a packet large enough to contain one would often be fragmented by the network layer, causing the loss of one fragment to render the others useless. They would also be packed unevenly into packets, resulting in a high overhead for packets containing small GOB fragments. The much smaller macroblocks can be packed efficiently into packets of around 1Kb, small enough to avoid fragmentation at the transport or network layers. Full decoder state is included at the start of each packet, making every undamaged packet useful to the receiver.

While Intra-H.261 loses the compression gains resulting from temporal prediction in other coding algorithms, at packet losses higher than 8% it outperforms even the newer H.263, and is more resilient to further losses. It is less complex to code and run, as only a small number of blocks are updated in each frame and the encoder does not need to run a partial copy of the decoder in order to form a prediction signal. This reduces the coupling required between sender and receiver and allows graceful degradation of the decoding algorithm by throwing away redundant updates. Both are important with the heterogeneous set of receivers common in a multicast session. Finally, Intra-H.261 takes advantage of the fact that lost block updates will typically be made quickly redundant by a new block transmitted due to continued change in that area. This is particularly true of the talking head in front of a large static background common in Internet seminars and conferences [McCanne96b].

3.4.3 Whiteboard

wb communicates between clients in the same session by multicasting commands in UDP packets. Each packet contains the source's identity, a timestamp, and one of 12 structured drawing commands. These range from RECTANGLE and LINE to Postscript wrappers that allow entire Postscript documents to be shared [Rasmusson95].

Because multicast runs over UDP, it does not provide end-to-end reliability in the manner of TCP. Moreover, for a number of reasons, this service is very difficult to provide. A TCP-style approach, where the sender continues to send a packet until an acknowledgement is received from each recipient, would be grossly inefficient and risk an ACK implosion effect [Erramilli87] where the large number of ACKs from each receiver overload the link back to the source. It would also require the sender to keep track of the state of each receiver, not a trivial task even if the group membership is available. Finally, adaptive algorithms such as TCP's congestion control can completely break down under conditions such as two orders of magnitude difference between the fastest and slowest links in a group [Pingali97]. Much research is in progress to solve these problems.

wb uses Scalable Reliable Multicast [Floyd95] to address these difficulties. SRM is an application-layer protocol that uses end-to-end mechanisms, making every recipient in a group co-operate to provide reliability to the other group members. Receivers detect loss through a gap in the sequence space used to label each packet or messages from other recipients detailing the latest received object. They wait a random time based on the distance from the node that triggered a request and multicast a request for the missing object. The other members who receive this request again wait for a random interval based on their distance from its sender, then multicast the requested data. If a host sees that another host has

requested information it is also missing, or that another host has already answered a repair request, then it will suppress its own response.

The total bandwidth used by the algorithm can be reduced by limiting the distance requests and repairs travel. This is particularly effective when small network areas are experiencing persistent loss, or when isolated newcomers request all the previous objects sent in a session.

SRM provides a fast and effective reliable multicast mechanism. It allows any group member to retransmit missing data, reducing latency and network core load. But it requires complex host-based mechanisms to keep the bandwidth required to an acceptable level.

3.5 Session descriptions, announcements and invitations

Session descriptions provide an advertisement that a session will exist, and also provide sufficient information including multicast addresses, ports, media formats and session times so that a receiver of the session description can join the session. The Session Description Protocol (SDP) specifies the content and format of this information [Handley98a].

One method of announcing sessions is to regularly send the session description on a well-known multicast address, with a specific scope, using the Session Announcement Protocol (SAP) [Handley99a]. The announcement includes information such as the organiser of the conference, and the session description. The bandwidth consumed by all announcements is limited, so senders must limit the rate that announcements are sent based on the fraction of the total bandwidth taken up by current announcements.

People wishing to participate in a particular conference must listen for the SAP announcement, and start up their tools with the SDP details provided. Session Directory (SDR) is a tool that can receive these “broadcast” session descriptions, browse through all sessions currently being announced, and then start up the relevant media viewers.

An important feature of SAP is that if the announcement of the message is received, there is a high probability that the session itself can be joined. The distance multicast packets travel can be restricted using the Time-To-Live (TTL) field in packet headers. The TTL is set by the sender and then decreased by each router through which it passes. Once the TTL reaches zero, routers no longer forward the packet. If the TTL of an announcement is set to be the same as the conference data, recipients who receive the announcement should also be able to receive the conference data. This also applies to regions with limited multicast connectivity to the sender of the data.

Another feature is that each SAP client sees all global SAP announcements; for this reason if it is announcing a new session at a particular time, it can check that there has not been a previous announcement for the same or another multicast group at the same time. This provides a primitive form of resource reservation.

Session announcements can also be used to advertise tightly-coupled sessions, which only requires that additional information about the mechanism used to join the session be given. However, as the number of sessions in the session directory grows, only larger-scale public sessions are likely to be announced in this manner; smaller, more private, sessions will tend to use direct invitation rather than advertisement. Otherwise either the bandwidth required by SAP, or the interval between announcements, will become too large.

The Session Initiation Protocol (SIP) provides an alternate mechanism whereby a user can be directly invited to participate in a conference [Handley99b]. SIP can be used whether a session is already ongoing, or is just being created. It is immaterial whether the conference is a small tightly coupled session or a huge broadcast – SIP merely conveys an invitation to a user in a timely manner, inviting her to participate, and provides enough information for her to be able to know what sort of session to expect. Thus although SIP can be used to make telephone-style calls, it is by no means restricted to that style of conference.

As many users are mobile, it is important that such an invitation mechanism be capable of locating and inviting a user in a location-independent manner. Thus user addresses need to be used as a level of indirection rather than for routing a call to a specific terminal. The invitation mechanism also provides for alternative responses, such as leaving a message or being referred to another user.

It is also possible to use off-line mechanisms for providing information on upcoming sessions. One is to send the information by e-mail; mailer plug-ins can parse messages and start sessions automatically [Hinsch96].

Alternately information can be put into a depository known to the potential participants. This mechanism is convenient if potential participants can be expected to access the directory sufficiently often. A combination of the use of ordinary e-mail to announce the existence of a conference, together with a directory mechanism to update session information, will probably be the most popular for a large class of applications. Specialised Web-based tools already allow browsing through lists of conferences, together with client plug-ins that can extract the relevant session description and start up the required tools.

Multicast sessions can be restricted using the TTL field or administrative scoping; hence although an announcement may be retrieved from a directory, it does not follow that the retriever can participate in a particular session. The directory server will often be at a different location than a session's sources, and the unicast route used to retrieve an announcement may be different from the multicast route session data will take.

Moreover, because the announcement is only seen by announcers using that directory, there is a chance of two sessions being announced for the same multicast address at the same time. By using random generation of multicast addresses, this danger can be reduced; however if the number of simultaneous sessions increase, the danger of such conflicts also increases. Requiring the use of a depository can also lead to it becoming a single point of failure. It is possible to ameliorate this problem by putting the announcements in several depositories; this is comparable with caching Domain Name System information in nameservers throughout the Internet.

None of these mechanisms will scale well to very large conferences, because of the potential number of messages or depository accesses. However it is not yet clear when tools will be available to manage really large conferences.

3.6 Security considerations in multicast conferencing

There is a temptation to believe that multicast is inherently less private than unicast communication since the traffic visits so many more places in the network. In fact, this is only the case with broadcast and prune-type multicast routing protocols such as Distance Vector Multicast Routing Protocol that forward a group's traffic everywhere that does not explicitly reject it [Deering91]. However, IP multicast does make it simple for a host to join a multicast group anonymously; it may then receive traffic destined to that group without the other senders' and receivers' knowledge. If the application requirement (conference policy) is to communicate between some defined set of users, then strict privacy can only be enforced through adequate end-to-end encryption.

RTP allows encryption of RTP and RTCP packets using symmetric algorithms. It specifies the US Data Encryption Standard (DES) [NIST88] but also allows the use of more secure ciphers. RTP also specifies a mechanism to manipulate passwords using the MD5 hash algorithm [Rivest92] so that the resulting bit string can be used as an encryption key. Passwords can be entered individually into the media tools, but it is more convenient for an application like SDR to provide the session encryption keys needed to encrypt and decrypt the media along with other tool parameters.

If IPSEC were to be used in place of application-level security, then the calls to launch the tools would only change slightly: the media encryption key would no longer need to be

supplied. The description would still set up the parameters of the tool; the encryption key would instead be inserted by SDR in the local Security Association Database, causing IPSEC to decrypt incoming and encrypt outgoing packets. Once standard IPSEC Application Programming Interfaces have been defined each tool could perform this function as it joins the relevant multicast group, removing the need for SDR to run as a trusted application.

The use of IPSEC in multicast environments has not yet been completely defined. In particular, there is no standard way to provide the keys needed to decrypt and authenticate packets to the authorised members of a multicast group. A number of proposals have been made, such as using secure point-to-point connections between a multicast group member and group controller to distribute keys [Canetti99], but are currently far from implementation. Controllers can assign unique Security Parameters Index values for a group to avoid collisions on values chosen by receivers as is standard with unicast IPSEC. If individual authentication or replay protection is required, the controller can also assign unique Security Association numbers to each group member.

Most current IPSEC implementations concentrate on securing streams at the host level, while conferencing is more concerned with the identity of persons participating. This gap is bridged by the mechanisms used for providing the keys used by IPSEC. For conferencing, these mechanisms should use personal authentication.

Data authentication in IPSEC uses a secret key shared between participants. While this will allow a group member to be sure data came from another member of the group, it does not identify which one. Public-key signatures would allow identification of individual participants, but are too expensive computationally to produce for any moderate stream rate. This problem seems best solved using hybrid signature schemes such as Rohatgi's that also use hash chains to allow low latency signatures, important for interactive multimedia [Rohatgi99].

A potential problem is that IPSEC is designed to operate at the IP level, while media addresses use both IP and port number. It may be desirable to use different cryptographic operations on different conference streams. With the present specifications, it will be necessary to have any streams with different security policies use different IP addresses. It is not yet clear whether this will cause operational difficulties.

Until multicast key negotiation mechanisms are standardised, setting up secured sessions will remain an application-level function. Security parameters are distributed in a session description using any of the methods described below. A tool such as SDR will insert security parameters from the session description into the local Security Association Database, causing traffic sent to and from that multicast address to be cryptographically processed using the standard IPSEC mechanisms.

3.7 Encrypted and authenticated session descriptions

Announcement and invitation to conference sessions is dependent on passing the session description to the authorised invitees. This information can be passed by many technologies – both in-band and out-of-band.

It is important that announcements and invitations can be authenticated. If making a network reservation results in usage based billing, then some form of authentication is essential. One may wish to be sure that someone who is authorised to do so has called the conference – important where a multicast address allocation scheme is in use. Finally, there are also mechanisms for modifying session announcements; a simple Denial-of-Service attack is to modify the announced time or location with unauthenticated announcements.

3.7.1 Authentication of session announcements

Anyone wishing to announce a conference can send out a session announcement. He signs the announcement, allowing it to be verified by recipients. It is impossible to verify his identity

without a certification infrastructure, but the main threat is a Denial of Service attack arising from an announcement being contradicted by a subsequent change in another announcement. Even without a security infrastructure, any recipient can check that the changed, received announcement came from the same source as the original announcement. It is not necessary to maintain an infrastructure to ensure that only the original announcer can alter a session.

3.7.2 Distributing session descriptions securely

A number of mechanisms can be used for distributing session descriptions. One is secure e-mail; S-MIME [Ramsdell99] or Pretty Good Privacy [Callas98] are the most commonly used message systems. The IETF conferencing community has a strong preference for PGP. A second is secure access to a Web depository, where mechanisms like TLS can be used. Figure 3.2 shows a prototype Web server we have been involved in developing that provides authenticated access to session descriptions. A third is access to a X.500 directory; here access control with strong authentication and encrypted transmission is appropriate. In these last two, an Access Control List, using either a password or a public key certificate, is used.

Although the secure DNS (DNSSEC) is appropriate for storing public keys, it is not an appropriate secured depository due to a design decision that there will be no control on access to DNS databases.

None of these mechanisms provide for changing keys during a session as might be required in some tightly coupled sessions, but they are sufficient for most usage in the context of lightweight sessions.

Figure 3.2: Secure Web access to session descriptions

Security extensions have also been developed for SIP that simply encrypt invitations to users with each user's public key and sign them with the sender's private key. For small conferences, each participant could just be invited individually.

It is important that the same session description can be used irrespective of the manner it is transferred. This allows the facility that launches the encrypted media tools to be oblivious of how the description came to the recipient.

Secure distribution is closely tied to authentication of group members. Conference or session keys can be distributed securely using public-key cryptography on a one-to-one basis. This security is only as good as the certification mechanism used to ensure the conference organiser holds the correct public key for each user. Such mechanisms are not, however,

specific to conferencing. Both X.509 [CCITT88] and PGP [Callas98] certificates can be used for this purpose.

Certificates may be revoked for many reasons; some will result from changes in roles, but others may be due to private key compromise or other security breaches. Certificate Revocation Lists therefore need to be checked regularly and certificate databases kept up to date.

A further complexity is that people "on the move" may miss session updates sent by e-mail but hidden in a large number of messages. Even with multicast accessibility at their current location, it may be difficult for them to be sure they have up-to-date session information. Clearly it is possible to provide additional plug-ins for mail systems that parse incoming messages for the MIME type that represents a description.

3.7.3 Use of encryption mechanisms with SAP

A level of indirection can be introduced by multicasting SAP messages encrypted using keys previously supplied to authorised recipients. This may be suitable for stable groups of individuals who hold regular conferences.

There is considerable controversy whether session announcements are an appropriate mechanism for announcing private sessions and thus whether there is a place for encrypted announcements at all. The bandwidth taken up by announcements is already quite large. It is typically some 10 minutes before an announcement is refreshed. Another concern is that one of the functions of session announcements is to avoid conflicts in the use of multicast addresses; this avoidance is impractical if the whole announcement is encrypted.

The first concern could be addressed by using session announcement cache proxies [Swan98]; these active services re-broadcast announcements at a higher rate over local subnets without needing access to the plaintext of encrypted announcements. The second concern could be met by separating out the address allocation functionality from the rest of the announcement mechanism. Both questions are still being discussed in the IETF. The use of proxies will not affect the time taken to refresh the proxies; it will, however, impact dramatically on the time for a workstation to obtain session descriptions from such proxies. But it is still not clear if the bandwidth saving of multicasting an encrypted announcement and separately supplying decryption keys is worth the complexity introduced, particularly regarding group changes that require new keys to be supplied to group members.

3.7.4 Changing encryption keys during sessions

A number of schemes are being investigated to change encryption keys during sessions. In the tightly-coupled sessions of H.323, these are included in the security standard [ITU98]. With multicast conferencing, it is possible to use different techniques – depending on whether the main threat is key compromise due to excessive use, or to a desire to change the set of participants. The first of these can be remedied by sending a new encryption key to the participants encrypted with the old key; at the same time the Session Announcement will need to be updated with the new key to accommodate late joiners. The second can be achieved only by out-of-band techniques – though some can still use largely the existing multicast trees [Wallner98]. None of these techniques have yet been standardised; they will not be considered further here.

3.8 Use of smart cards for secure conferencing

All the security mechanisms described so far rely upon each conference participant keeping secret the keys necessary to decrypt session descriptions and encrypted media streams. Most current software stores this information on an individual's workstation, usually in encrypted form. It is possible to streamline many of the operational issues if keys are instead kept on a smartcard. Such a card is easily portable, and can be used on other workstations without the

need to transfer keys in software. Secunet[1] has already demonstrated an IPSEC stack that can use a smartcard to decrypt and authenticate data.

It is expected that use of such smartcards will greatly ease the operational deployment of secure conferencing. It should be much easier to automate operations, and to reduce the security constraints on the workstations, with the use of such cards. Users are also relieved of the burden of managing private keys, particularly remembering long passphrases.

3.9 Conclusion

The critical requirement for a secure conference is end-to-end encryption of its media streams. We have described how to use application and network-layer cryptography to achieve this goal, and the relative merits of each.

We have also discussed the more difficult problem of distributing the session keys used to authorised participants. Until the IETF standardises a method of multicast key distribution, encrypted announcements and invitations and secure e-mail and depository access are perfectly adequate for the needs of lightweight conferences.

This shows a practical application of our contention in chapter two that a combination of network and application-layer security allows flexible and efficient communications security over the Internet. We will show in chapters five to seven that it also allows active network services to operate on those communications without requiring their security to be compromised.

[1] http://www.secunet.de/

4 Unconventional threats to active services

"To address the emerging cyber threat, the [Department of Justice] FY 2002 budget includes $33 million in increased resources. Within this amount, $28.14 million will support the FBI's counter-encryption capabilities, and the development of cyber technologies for the interception and management of digital evidence."

–US Attorney-General John Ashcroft

"We will firewall Napster at source – we will block it at your cable company, we will block it at your phone company, we will block it at your ISP. We will firewall it at your PC."

–Sony Senior VP Steve Heckler

"We shall not flag or fail. We shall go on to the end. We shall fight in France, we shall fight on the seas and oceans, we shall fight with growing confidence and growing strength in the air, we shall defend our island, whatever the cost may be. We shall fight on the beaches, we shall fight on the landing grounds, we shall fight in the fields and in the streets, we shall fight in the hills; we shall never surrender."

–Winston Churchill

4.1 Introduction

Many active service schemes rely on access to recipients' cryptographic keys or the plaintext of data they wish to process. The general-purpose PCs running the majority of these services are completely unsuitable for this purpose due to their lack of security. We will merely agree with the US National Security Agency [Loscocco98] on how serious this deficiency is – the `comp.risks` newsgroup provides a good introduction to the voluminous and ever-growing literature on problems in system security. It is unfortunate that the networking and security research communities are so separate that few active service designers realise the problems this can cause.

The demands for serious security from sectors such as the financial services industry has led to the development of hardened nodes to run these services [nCipher00]. But we describe in this chapter why this "fix" comes at the problem from the wrong direction. There will be continuing access to supposedly protected data in those nodes; and this will lead to legal problems in systems whose designers do not realise the high legal tests that concepts such as "non-repudiation" must meet in the real world.

In this chapter we describe "unconventional" threats, mostly ignored in the security literature, which should be included in any active service security model that includes real-world attacks. We examine the misuse of the judicial discovery process, Customs powers, newly emerging key disclosure warrants, and misdirected signals intelligence product. Old-fashioned intimidation and blackmail are also considered. Principals are vulnerable not just to repudiation of contracts and instructions, but to economic intelligence gathering, breach of due care obligations such as under non-disclosure agreements, revelation of trade secrets, or even the exposure of the identity of whistleblowers. And the 'common carrier' status of network operators may come under threat if they are able to police their users on behalf of government and corporate interests.

We also examine the problems that can occur when active services manipulate security information that is supposed to provide non-repudiation in protocols. We show that the terms and conditions in many electronic payment systems leave consumers at risk of sustaining serious financial losses, and that this could be a major problem for "secure" active services such as WAP gateways [WAP98] that are already being used by banks to enable mobile account management by their customers.

4.2 Unconventional threats

4.2.1 Judicial discovery processes

Courts the world over have extensive powers to order the production of information by parties to a case. Witnesses can be compelled to give evidence and produce documents in intelligible form. Non-compliance can lead to criminal contempt of court charges, or loss of the case. US litigation in particular relies heavily on a pre-trial discovery process where the parties may judge the strength of their case by viewing their opponents' evidence. Active service providers with access to plaintext of data of interest to a court may expect subpoenas from both parties demanding copies of that information.

The anti-trust case against Microsoft has shown how devastating discovery powers can be. The initial Justice Department complaint relied heavily on internal Microsoft e-mail obtained during their investigation. Executive after executive, all the way up to Bill Gates, was extensively quoted describing how Windows should be leveraged to beat Netscape in the browser wars. Microsoft's Christian Wildfeuer, for example, allegedly wrote that "It seems clear that it will be very hard to increase browser market share on the merits of [Internet Explorer] 4 alone. It will be more important to leverage the OS asset to make people use IE instead of Navigator" [Goodin98]. Faced with such evidence, it seems unsurprising that Judge Thomas Penfield Jackson eventually found that Microsoft had abused their dominant position in the operating system market.

The Fourth and Fifth Amendments to the US Constitution provide limited protection against search and seizure and compelled self-incrimination for individuals. But a series of Supreme Court judgments have made this protection limited indeed in the case of information existing in physical form [Sergienko96]. Company documents are explicitly exempt: "The [Fifth] amendment is limited to a person who shall be compelled in any criminal case to be a witness against himself; and if he cannot set up the privilege of a third person, he certainly cannot set up the privilege of a corporation" [US06]. And only when a cryptographic key is memorised rather than written down is an individual protected if its disclosure would provide 'testimonial' evidence of criminal activity [Sergienko96].

Groups have used these processes to discover, among other things, the identity of pseudonymous on-line critics. The Church of Scientology subpoened the e-mail address behind a pseudonym at `penet.fi`, one of the original anonymising remailers. Unfortunately for them, it pointed only to another pseudonym at a more advanced remailer at `c2.net` [McCullough96]. ITEX Corporation sued Yahoo to discover the identity of 100 "John Doe" critics who had made negative comments on a Yahoo Finance message board [Macavinta99]. In neither case were the individuals behind the pseudonyms given the opportunity to first present the case against their exposure.

These pressures, combined with data protection laws in many countries, will encourage companies to limit the amount of e-mail they retain for any amount of time. Regulators require certain classes of information to be kept for varying periods of time. The US Securities and Exchange Commission, for example, has ruled that brokers must keep transaction and participant information for automated trading systems for three years [SEC95]. Lawyers suggest that otherwise documents should generally be kept one year beyond the local relevant statute of limitations [Carroll98].

There is often little reason why data should be kept for one moment longer. Companies should have procedures in place to allow the effective destruction of all copies of a document, including backups. These policies should be followed as strictly as possible, as selective document destruction may lead investigators to assume specific incriminating information has been purged [Simmons97].

E-mail destruction policies are especially important. Because users tend to regard messages as closer to telephone conversations than letters, they metaphorically commit to paper

information that should never have been given permanence. Copies of indiscrete messages on disks, backup tapes, or mail servers can threaten the very existence of a company, as the Microsoft case has shown. Even fragments of old messages that have been insecurely deleted or left in swapfiles are vulnerable. Under the UK Criminal Justice and Police Act 2001 s.50(1), "where a person who is lawfully on any premises finds anything on those premises that he has reasonable grounds for believing may be or may contain something, for which he is authorised to search on those premises… that person's powers of seizure shall include power under this section to seize so much of what he has found as it is necessary to remove from the premises to enable that to be determined." This could include a machine running an active service, even if its loss would damage other elements of the operator's system.

An e-mail discovery request can also prove extremely expensive for companies with large collections of old messages. Computer Forensics, Inc. estimates that "the process of reviewing and collecting e-mail from disparate tapes and databases usually costs at least $25,000" [Fusaro98].

Companies implementing data destruction policies will be most concerned about active service operators with plaintext access that may be compelled to provide copies of that data to hostile parties outside the scope of their destruction policies.

4.2.2 Import and export searches

Customs authorities have very wide-ranging powers to search materials being imported to and exported from a country. They have recently used these powers to investigate digital information such as that on laptop hard disks. UK Customs and Excise have declared that officials will routinely scan laptops for illegal material such as pornography [Davies98] without any requirement for probable cause as exists in the US. Journalist Ken Cukier reported that Customs randomly attempted to scan his laptop on arriving in London on Eurostar (but were thwarted due to his use of an Apple machine) [Nuttal98]. Whether scans will also be used to further the "economic well-being" of the UK must be judged by those carrying information that would help that aim. W. H. Murray, an information security consultant with Deloitte and Touche, concluded:

> "While I may not have anything on my laptop that C&E have any legitimate objection to, I have much on my laptop that is none of HM's government's legitimate business. Some of it is personal. Much of it is data of or about my clients which I have a professional, ethical, and contractual obligation to keep confidential. Some of it might never have been shared if such procedures had been routine, to my detriment, to that of my clients, and to that of the commonwealth... I will have to advise my clients to eschew discretionary travel through or to a country that has such procedures. Since I never travel without my computer, would not be much value to my clients without it, and carry on it information which I owe a duty to my clients to protect from copying, I will not go back to England unless and until HM's government renounces such extreme measures. As it is one of my favorite destinations, I do not say that lightly." [Murray98]

It is absolutely inevitable that Customs services will turn their attention to information being transferred internationally in electronic form as travellers start restricting the data held physically on their laptops' hard disks. And given that much domestic Internet traffic is routed internationally due to network configuration, a large percentage of traffic may become liable to examination. Any active service operator with access to the plaintext of international traffic that Customs finds interesting will receive demands for copies to be provided.

4.2.3 Decryption and key warrants

The Organisation for Economic Cooperation and Development guidelines on encryption state that "National cryptography policies may allow lawful access to plaintext, or cryptographic keys, of encrypted data." [OECD97] A number of governments have considered legislation that would require plaintext or keys to be provided by an individual or organisation under warrant. So far the UK, India, Singapore and Malaysia have implemented such legislation. The US, Belgium and Netherlands have acts pending that require third parties in possession of keys to provide them to authorities [Banisar00]. The draft Council of Europe Convention on Cyber-Crime [CoE01] requires signatories to provide legislative powers to force individuals to reveal any reasonable information required to search or copy seized secure data (article 19(4)).

The Regulation of Investigatory Powers Act 2000 provides these powers in the UK. Its decryption or key request notices may also contain a gagging clause that carries five years' imprisonment for an individual notifying anyone other than their lawyer that the notice has been served. The UK Home Secretary may authorise notices "for the purpose of safeguarding the economic well-being of the United Kingdom" (s.5(3)(c)). And notices may be served "for the purpose of securing the effective exercise or proper performance by any public authority of any statutory power or statutory duty" (s.49(2)(b)(ii)) – a *huge* number of bodies in the UK. Serious doubts have been raised about the ability of public authorities to provide security for disclosed keys commensurate with their value [Gladman00]. The Act requires that communications service providers with the ability to decrypt specified data do so upon receipt of a decryption notice (s.49(2)(a)). They may also, given a proportionality test is met, have to provide copies of longer-term keys resident within their system. It has been estimated that the resulting ISP costs and reduction of consumer and business confidence in system security could cost the UK economy £46bn over the next five years [Brown00b].

Any keys used to protect traffic and instructions to active services will be vulnerable to seizure under these powers.

4.2.4 Signals Intelligence

Perhaps the most pervasive threat to confidential information is the activities of the world's Signals Intelligence (SIGINT) agencies. These government organisations use a vast array of technologies to capture communications from commercial satellites, long distance communications, undersea cables, and at many points on the Internet. More than 120 satellites are in operation to support their activities. Members of the five-nation UKUSA alliance (the US, UK, Canada, Australia and New Zealand) share SIGINT facilities, tasks and product [Campbell99].

While the National Security Agency, Britain's Government Communications Headquarters, and their many foreign equivalents have become slightly better known during the past twenty years, few outside the security community realise the vast scale of their activities. The 1998 NSA budget is estimated at $3.6bn, with significant further costs incurred in the $6.3bn National Reconnaisance Office expenditures [Pike98]. Globally, it is estimated that 15—20bn is spent every year on communications intelligence [Campbell99].

The US Foreign Intelligence Advisory Board recommended in 1970 that "henceforth economic intelligence be considered a function of the national security, enjoying a priority equivalent to diplomatic, military, technological intelligence" [Campbell93]. The National Security Agency is authorised by Executive Order 12333 to collect "information for national foreign intelligence purposes in accordance with guidance from the Director of Central Intelligence" (s.1.12(b)(3)) [US81]. The secret Office of Intelligence Liason, renamed in 1993 to the Office of Executive Support, routes intelligence information to the Department of Commerce, from where "tips based on spying ... regularly flow from the Commerce Department to U.S. companies to help them win contracts overseas" [Shane96]. A report from

the European Parliament states that deals worth billions and billions of dollars have been won by US companies after receiving signals intelligence from their government [Campbell99].

While claiming US economic espionage is limited to preventing bribery winning contracts, ex-Director of Central Intelligence James Woolsey admitted that "We steal [economic] secrets with espionage, with communications, with reconnaissance satellites" [Woolsey00]. And letters from the CIA to Congress disclose that the intelligence agencies are also interested in "lobbying," "linking financial aid to contract awards" and "the use of insider information and disinformation against U.S. firms." The CIA's National Counter Intelligence Center reported to Congress that "because they are so easily accessed and intercepted, corporate telecommunications – particularly international telecommunications – provide a highly vulnerable and lucrative source for anyone interested in obtaining trade secrets or competitive information." [Windrem00]

GCHQ provides similar economic intelligence in the UK. It is authorised by the Intelligence Services Act 1994 to intercept foreign communications "in the interests of the economic well-being of the United Kingdom" (s.3(2)(b)). Targets may be specified by the government's Overseas Economic Intelligence Committee, the Economic Section of the Joint Intelligence Committee, the Treasury, or the Bank of England [Urban96]. MI5, Britain's internal Security Service, is spending £25m on a National Technical Assistance Centre at its London headquarters to monitor traffic and attempt to decrypt captured ciphertext [Rufford00].

These agencies are moving to require that modern networks are as surveillance-friendly as the monolithic state-owned monopolies they are replacing. The US Communications Assistance to Law Enforcement Act 1994 and UK Regulation of Investigatory Powers Act 2000 both mandate that network operators include wiretap capabilities within their systems. CALEA specifies that "a telecommunications carrier shall ensure that its equipment, facilities, or services… are capable of… expeditiously isolating and enabling the government, pursuant to a court order or other lawful authorization, to intercept, to the exclusion of any other communications, all wire and electronic communications carried by the carrier within a service area to or from equipment, facilities, or services of a subscriber of such carrier concurrently with their transmission to or from the subscriber's equipment, facility, or service" (s.103(a)(1)). The earlier Foreign Intelligence Surveillance Act 1978 allows the Attorney General to direct a common carrier to "furnish all information, facilities, or technical assistance necessary to accomplish the electronic surveillance in such a manner as will protect its secrecy" (s.1802(4)(A)). CALEA allows a civil court to impose a penalty of up to $10,000 per day for violation of orders upon "a telecommunications carrier, a manufacturer of telecommunications transmission or switching equipment, or a provider of telecommunications support services" (s.2522(c)(1)). And revealing classified information "concerning the communication intelligence activities of the United States or any foreign government" is punishable by up to ten years in jail (s.798(a)).

The RIP Act gives the UK government even wider latitude and powers. "The Secretary of State may by order provide for the imposition by him… of such obligations as it appears to him reasonable to impose for the purpose of securing that it is and remains practicable for requirements to provide assistance in relation to interception warrants to be imposed and complied with" (s.12(1)). The Secretary of State may issue interception warrants "in the interests of national security; for the purpose of preventing or detecting serious crime; for the purpose of safeguarding the economic well-being of the United Kingdom" (s.5(3)(a—c)) where "serious crime" is later defined to include any criminal conduct "by a large number of persons in pursuit of a common purpose" (s.81(3)(b)). These warrants are addressed to one of the directors of MI5, MI6, GCHQ, NCIS, Defence Intelligence, any chief constable or the Comissioners of Customs and Excise (s.6(2)), and "shall be taken to include… conduct by any person which is conduct in pursuance of a requirement imposed by or on behalf of the person to whom the warrant is addressed to be provided with assistance with giving effect to the warrant" (s.5(6)(c)). A special "certificated warrant" issued under s.8(4) and s.15(3) appears to allow GCHQ to conduct mass trawls through domestic traffic [Bowden00].

Both nations allow tapping by executive fiat. “The President, through the Attorney General, may authorize electronic surveillance without a court order under this subchapter to acquire foreign intelligence information for periods of up to one year… between or among foreign powers” (FISA s.1802(a)(1)). RIP warrants are always Government-issued, and subject to oversight only by an Interception Commissioner appointed by and reporting to the Prime Minister. The UK Parliamentary Intelligence Select Committee found that in 1999, surveillance oversight was “dependent on a tiny support structure which is quite incapable of carrying out the job. As we reported, there was not even anybody to open the mail, let alone process it, for many months. That was ludicrous” [Beith01].

Telecommunications companies are working hard to provide these intercept capabilities efficiently. The European Telecommunications Standards Institute has a whole series of documents on the requirements of and methods for providing interception. Each of its technology bodies is “responsible for the production of a technology specific mapping of syntax and behaviour to meet the requirements established” [Cadzow01b]. An ASN.1 syntax and set of required behaviour is defined for the transfer of intercepted information to a Law Enforcement Monitoring Facility [ETSI99]. And network operators are prohibited from optimising the performance of calls where interception would be affected: “When a call set-up involving a notified target is received at a Network Functional Group it shall refuse any request to permit a shorter media path which may be requested by other Functional Groups in the call” [Cadzow01b].

CALEA and RIP both require that carriers must decrypt any traffic where they possess the necessary cryptographic keys. CALEA states that “A telecommunications carrier shall not be responsible for decrypting, or ensuring the government's ability to decrypt, any communication encrypted by a subscriber or customer, unless the encryption was provided by the carrier and the carrier possesses the information necessary to decrypt the communication” (s.103(b)(3)) and RIP allows ‘disclosure notices’ to be served when “a key to the protected information is in the possession of any person” (s.49(2)(a)).

These powers would already seem sufficient to require that the operator of an active service with access to plaintext of data could be forced to secretly provide copies of that data to a wide range of governmental bodies through automated interfaces. David Herson, an ex-member of GCHQ and the EU’s Senior Officers Group on Information Security, candidly admitted in 1996 that “Law Enforcement is a protective shield for all the other governmental activities. You should use the right word - we're talking about foreign intelligence, that's what we're talking about - that's what all this is about. There is no question - that's what it is about. The Law enforcement is a smoke screen” [Nielson96].

4.2.5 Insider attacks

The oldest threat against security systems is corruption or coercion of authorised users. The strongest cipher is no protection if a key or plaintext can be obtained from its owner. Two separate keyholders are required to open bank vaults as much to protect bank managers’ families against kidnapping as to protect against corrupt staff [Anderson01].

Company employees are well-placed to steal economic intelligence for competitors. Trade secrets can prove astonishingly valuable. Volkswagen and General Motors settled a trade secret theft case out of court for $100m and an agreement that Volkswagen would buy at least $1bn of GM parts [DN97]. An Intel software engineer was arrested for stealing and attempting to resell $20m of company secrets [FBI97]. Cisco estimated that a stolen copy of its Private Internet Exchange software was worth $2bn [Romano00]. An Ernst and Young survey found that 82% of the worst frauds suffered by respondents were due to employees [Sherwin00].

Active service designers must remember the vulnerabilities introduced by administrators who may have guns held to their heads, be entrapped by sexually attractive conspirators, or be

subject to a myriad of attacks dreamt up by imaginative criminals and high-tech thriller authors!

4.2.6 Loss of common carrier status

Internet Service Providers are largely protected from liability for the content they carry over their networks under the 'common carrier' doctrine that has long protected telecommunications companies. But recent court decisions have shown that this protection could easily be lost if ISPs have the ability to police that content.

The US film and music industries have taken a particularly aggressive lead in demanding the cooperation of service providers in policing the action of their users. The federal appeals court decision over use of the Napster file-swapping software held that:

> "if a computer system operator learns of specific infringing material available on his system and fails to purge such material from the system, the operator knows of and contributes to direct infringement… Conversely, absent any specific information which identifies infringing activity, a computer system operator cannot be liable for contributory infringement merely because the structure of the system allows for the exchange of copyrighted material… sufficient knowledge exists to impose contributory liability when linked to demonstrated infringing use of the Napster system… Napster has actual knowledge that specific infringing material is available using its system, that it could block access to the system by suppliers of the infringing material, and that it failed to remove the material" [Schroeder01].

The court further found that Napster's ability to regulate the use of its system meant that by not doing so, it was vicariously liable for copyright infringement. "Turning a blind eye to detectable acts of infringement for the sake of profit gives rise to liability… Napster's reserved 'right and ability' to police is cabined by the system's current architecture."

This determination could prove costly for Napster. A previous judgement against MP3.com awarded $25,000 per CD copied in damages, estimating a total of around $118m [Rakoff00]; the Napster plaintiffs have asked for $100,000 per work infringed as well as attorneys' fees, restitution and punitive damages [Frackman00].

As replacement services start springing up out of the jurisdictional reach of the US courts, it is likely that the Recording Industry Association of America will attempt to use this judgement directly against ISPs and their users. It is not a huge leap of legal logic to decide that the contents of a Web cache, or the traffic to and from known Napster clone sites, are within the 'right and ability' of ISPs to police. Several record labels have demanded that ISPs block connections from users identified as making copyrighted material available [Borland01]. Microsystems Inc. has already served e-mail subpoenas demanding the contents of Web logs that list IP addresses that have downloaded copies of the `cphack` decryption program [Bridis00]. Belgian police raided the homes of three music-sharing site users in February 2001 "looking for evidence of copyright infringement" [AP01]. And Excite@Home Australia is already monitoring its network for evidence of the downloading of pirated files, immediately terminating the account of the responsible user [McAuliffe01].

As a major exporter of intellectual property, the US has taken a hard line on the development and enforcement of IP law by other nations. It pushed through a wide range of additional copy-rights in the World Intellectual Property Organisation's update to international copyright law, including restrictions on "infringing technologies" that can be used to circumvent copy protection mechanisms. The US administration was even accused of pulling an "end-run" around the US Congress by creating these strengthened powers by treaty then imposing them upon the national legislature [Samuelson96]. And indeed, the Digital Millenium Copyright Act 1998 brought these restrictions into US law. The Uniform Commercial Information Transactions Act, currently being pushed through state legislatures,

gives IP owners new rights to enforce information contracts against "licensees" of their property [Samuelson98, Samuelson99]; and the Hague Conference on Private International Law's Proposed Convention on Jurisdiction and Foreign Judgments in Civil and Commercial Matters would make these and other information rights enforceable in many cases throughout the 49 member nations of the Conference. This would place ISPs unwilling to block access to materials illegal in any of the member states at risk of asset seizure [Love01].

Meanwhile, many of the EU member states have been pursuing the extra-territorial enforcement of their "hate speech" statutes. A French court has ordered that California-based Yahoo ban its French customers from purchasing Nazi memorabilia [Hu00], despite the difficulties in identifying the geographic location of Internet users. A lawsuit aiming to require filtering of one US "hate-speech portal" has already been filed against 14 French ISPs [BNA01]. The European Parliament has called for service providers to be liable for "criminally unlawful content of third-party services provided by them… if they are expressly aware of the specific contents and if it is technically possible and reasonable for them to prevent their use" [EP97]. And the Council of Europe's Parliamentary Assembly has called for the Council's draft cybercrime treaty to include a protocol banning "racist propaganda" and "abusive storage of hateful messages" [Tallo01].

To avoid being forced to act as enforcement agents for a wide range of governmental and corporate interests, and attracting various forms of existing and future liability, ISPs should think carefully before deploying systems that give them the ability to police content flowing through their networks.

4.3 Real-life non-repudiation

Many public-key authentication algorithms claim to provide non-repudiation: a signature verified by a public key was provably made by the owner of the related private key, and cannot be disavowed. This is in contrast to symmetric authentication algorithms, where either party to a communication could have created the authenticator.

It may be mathematically probable (given the continuing lack of proof of security for asymmetric cryptosystems) that only someone in possession of the related private key can have made a given signature. But "cryptographic theorists often ignore a messy detail that lies between Alice and her key: her computer" [Ellison00]. The security of that key, and linking its use to the intent of its owner, are far greater practical obstacles to providing non-repudiation of signed data.

This becomes a problem when financial institutions start allowing customers to issue "secure" electronic instructions at their own risk. Many of the UK banks have terms and conditions containing provisions similar to that of the Halifax: "You will not be responsible for any transactions using your password or any of your additional security details after you have told us that they might be known or used by someone else" [Bohm00]. As the first indication many consumers might receive of misuse of their security details would be the loss of their entire account balance, it is of little consolation that they can then remove further liability by notifying the bank.

This is in stark contrast to the liability regime of previous financial instruments. Under the Bills of Exchange Act 1882, "…where a signature on a bill is forged ... the forged ... signature is wholly inoperative, and no right to retain the bill or to give a discharge therefor or to enforce payment thereof against any party thereto can be acquired through or under that signature" (s.24) – thus ensuring that banks rather than customers bear the risk of cheque forgery. Similarly, under the Consumer Credit Act 1974 and its regulations, customer losses are limited to £50: many banks waive even this liability [Bohm00].

The interception of security information or insertion of false instructions into an authenticated connection could therefore be disastrous for users of online banking. Yet one of the most heavily-promoted uses of the Wireless Application Protocol suite – with its use of a gateway

that decrypts, translates then re-encrypts data near the base station [WAP98] – has been on-line account management.

The security ambiguity this introduces could also create many problems for banks. In previous cases where customers and banks have argued in court over disputed debits on accounts, the banks have tried to assert that their computer systems are so secure that the customer must be attempting to commit fraud. This approach was initially successful: a policeman was convicted after disavowing a "phantom" ATM withdrawal, and subsequently lost his job. Upon appeal, the court ordered that the defence's expert witness be allowed to examine the bank systems: he found a large number of security flaws. The appeal was successful [Anderson94]. If banks wish to attract further bad publicity and detailed attention to their security systems, they should be wary of introducing such a major point of attack and hostage to fortune by expert witnesses.

Producing digital evidence that can stand up to hostile cross-examination has been a continuing problem for law enforcement agencies [Anderson94, Sommer98]. There are many different ways that doubt can be introduced even in the mind of magistrates [Davies00]; lay juries unlikely to contain computer specialists may feel it impossible to convict based on heavily disputed technical evidence. Digital signatures are one of the few available mechanisms that can allow the production of reasonably reliable evidence. It would be unfortunate if the use of active services negated that benefit.

4.4 Conclusion

Steve Deering has quipped that IPSEC should have been part of the IP protocol suite from the beginning so that active services could not have been developed in the presence of end-to-end encryption[2]. Given the unfortunate reality that this was not the case, most active service designers now feel that the benefits of their services outweigh those of end-to-end security.

We have attempted to demonstrate in this chapter that this is a high standard to meet. Even reaching the point where systems that run active services meet the very high standards of security required will be difficult. Related techno-legal research has suggested that the US Department of Defense's B1 rating should be an appropriate minimum for such systems [McCullagh00]. The scarcity and high cost of such systems does not bode well for their widespread use within networks.

But we suggest that so many threats would still remain – particularly those related to governments' voracious appetite for communications intelligence and corporations' determination to enforce intellectual property rights – that this would provide a false sense of security. Active service node operators that have the ability to examine plaintext traffic may become required to police that traffic. Any widespread use of secure nodes would quickly lead to the legal requirements for "law enforcement access" interfaces that are widespread in the telephony world. Whether used as intended or by attackers exploiting weaknesses in those interfaces, the security expected by the endpoints of a data transfer may be fatally compromised.

This would be particularly unfortunate given the faltering progress being made in providing evidential value to data. Digital signatures are one of the few available mechanisms that allow companies to depend on remote instructions given by customers. That dependence will become risky if courts start deciding that an instruction was just as likely to have come from a hacked intermediate point.

Obtaining "common carrier" protection against liability for content carried by telecommunications operators was a hard-won legal battle. If telephone calls were as easily monitored and policed as plaintext Internet data, this protection may never have been granted; but since it is available and would continue to be so where Internet service providers can

[2] Personal communication

demonstrate that they cannot access the content of ciphertext traversing their networks, it would seem unfortunate to lose it now.

Because all of these threats are so difficult to quantify, active services with plaintext access introduce an unmanageable risk that users and network operators may be unwilling to bear.

5 Distributed packet filtering

"The real problem causing today's DDOS attacks is that these systems (Windows and Linux) are so vulnerable to Trojans and other attacks that allow the distributed deployment of agents/Zombies... We have the law enforcement community to thank for blocking end-to-end security and authentication, and we have the OS designers to blame for not developing good sandboxing virtual machine capabilities"

–FCC Chief Technologist Prof. Dave Farber

"A brand new generation of zombies is lying in wait in the dark corners and unused basements of some computer networks, waiting to be released so that they can eat the brains of the Internet."

–Michelle Delio

5.1 Introduction

The Internet is made up of connections with very different Quality of Service characteristics and capabilities. Wireless links, now becoming increasingly popular, have far lower bandwidth and higher delay, jitter and error rates than wired connections. While hosts with high-speed fixed links have the bandwidth available to send and receive large amounts of multimedia data, it would completely swamp wireless links. The next generation of mobile telephones providing video links are likely to vastly outnumber the current number of devices connected to the Internet receiving video data.

This is a problem for systems sending shared data to multiple recipients over such links. They are forced down to a lowest common denominator approach, where a group containing five clients on T1 links are all forced to communicate at the rate of the client on the slow GSM connection; or to exclude clients that cannot reach the QoS expectations of other group members. Or they have to use a point-to-point topology, which scales badly and does not take advantage of the underlying network's ability to eliminate replicated data. This would be particularly inefficient on the Internet, which already has a multipeer connection mechanism in the form of multicast.

A number of mechanisms have been developed to retain the benefits of multicast whilst catering for the differing needs of heterogeneous clients. Receiver-driven Layered Multicast [McCanne96b] and Scalable Consensus-based Bandwidth Adaptation [Amir98a] are end-to-end schemes that allow different clients to vary the amount of data they receive and to specify to data sources how they should share the available bandwidth. Other schemes like the UCL Transcoding Gateway use active services that adapt data flows to the characteristics of connected links – in this case, reducing the bandwidth used by data travelling over a low-capacity link [Kirstein98]. Such schemes require plaintext access to perform processing such as transcoding between video formats.

This chapter analyses the features present and absent in these schemes, and then proposes a new protocol – congestion hints – that overcomes some of their problems and can work with encrypted traffic. It also shows how this mechanism can be used to combat flooding attacks such as the Distributed Denial of Service attacks seen in February 2000 [Todd00].

5.2 End-to-end mechanisms

5.2.1 Receiver-driven Layered Multicast

Layered multicast is a set of protocols that allow data to be distributed in several different layers that each provide increasing quality or speed. Special compression algorithms are used that allow data to be striped across different layers, which are then transmitted over different multicast groups. Clients subscribe to as many multicast groups as their available bandwidth

allows. Each additional layer received improves the quality of real-time data such as video [McCanne96b] or the speed at which best-effort traffic is received [Luby01].

Receivers 'probe' network capacity (in a similar way to TCP's congestion control mechanisms) by leaving groups when congestion is detected and experimenting with joining groups to increase throughput, in an effort to get closer to the optimal network utilisation. Clients wait for a *join-timer* to expire before attempting to join a given layer. A separate timer is maintained for each layer and multiplicatively increased when congestion is encountered. This means that clients will quickly add layers as they move towards the optimum bandwidth level, but then check for futher increased bandwidth infrequently.

To improve scalability, clients notify other group members when performing join-experiments using a multicast message. Other group members then wait for the results of the experiment before starting a new one of their own. The whole group learns from the results of those experiments by examining their effect on local congestion levels; each member updates their own join-timers appropriately [McCanne96a].

5.2.2 Scalable Consensus-based Bandwidth Adaptation

SCUBA is a protocol that allows data sources in a multicast session to determine the consensus of interest of receivers in their data. They can then alter the amount of data they transmit to reflect the group's consensus, fairly allocating the available bandwidth between senders [Amir98a].

SCUBA receivers periodically report their interest in each sender using a multicast RTCP message containing a vector of addresses and weights totalling one, as shown in figure 5.1. The overall bandwidth used by the protocol is constrained to some small fixed percentage of the channel's capacity. Senders estimate the average weight that should be assigned to their data by averaging the last *m* reports received. Larger values of *m* increase the accuracy of the estimate but also the time it takes for the group consensus to be determined, particularly if the variance of the receivers' weights is high.

Amir noted that humans rarely simultaneously attend to a large number of audio/video sources such as a conference participant. He suggested this could be used to limit the size of SCUBA reports by limiting the interest of each receiver to a small number of senders. Combined with estimating rather than calculating the group's consensus on source weights, these factors mean SCUBA can scale to a large number of members of a session.

Source address	Weight
128.16.5.135	0.75
128.16.5.134	0.15
128.16.5.133	0
128.16.5.132	0
128.16.5.131	0.10

Table 5.1: example SCUBA report

Amir calculated that with an average report size of 66 bytes[3], receiver $\sigma^2 = 0.1$ and messages limited to 5% of a 128kb/s channel, it would take five seconds for a reasonably accurate (α=0.1) estimate to be made of receivers' preferences.

SCUBA also allows the output of layered codecs to be rearranged across different multicast groups according to receiver interest. Rather than all sources sending signal layer *n* to the same network layer *n*, higher-priority sources use more of the lower network layers so that a

[3] 5 * 4 bytes per source ID and 2 bytes per weight plus 28 bytes for a UDP header and 8 bytes for a SCUBA header.

client subscribing to a limited number of layers receives more data from the more popular sources.

5.3 Media gateways

Media gateways are active services that allow a stream of media data multicast by a sender to be adapted as it travels through the network over links with different characteristics, most commonly reducing the bandwidth required. Gateways usually concentrate on video streams, as they contain a far greater amount of data than other media flows. The University of California, Berkeley's `vgw` (video gateway), for example, contains filters that can convert 6Mb/s Motion-JPEG data to 128kb/s H.261 data [Amir95].

Yeadon comprehensively surveyed available filtering mechanisms [Yeadon96], which can be broadly divided into the following groups:

Frame dropping: simple (drop late and partially corrupted frames, and a dynamic percentage of frames) and priority-based (drop less important frames such as B and P frames in an MPEG stream). Requires packets to be labelled with a frame ID and type.

Transcoding: convert between different compression or data standards. The Robust Audio Tool transcoder, for example, converts between the Pulse Code Modulation, DVI, Global System for Mobile communication and LPC formats [Kirstein98].

Colour reduction: remove/reduce chrominance but leave luminance. Monochrome or Discrete Cosine Transform-based. Has proven very effective.

Dithering: reduce bits-per-pixel.

Low-pass: remove high frequency components (AC coefficients above a cut-off frequency).

Re-quantisation: approximates each DCT-coefficient value.

Limiting: keeps each frame's data below a certain threshold by dynamically adjusting cut-off frequency and/or requantisation step size. This converts a bursty variable bit-rate stream into a constant bit-rate stream.

Splitting: separate a mixed media stream or hierarchically encode a stream, allowing specific QoS parameters for each stream. For example, conference audio data should have a higher priority than video; MPEG Bidirectional pictures should have a lower priority than Intra pictures.

The following graph [Yeadon96 p.121] shows the amount of data reduction achieved by a selection of filters on a test MPEG file. The filters operated quickly enough on standard PCs to service 2—4 output streams to machines of equivalent capability with no extra delay or jitter.

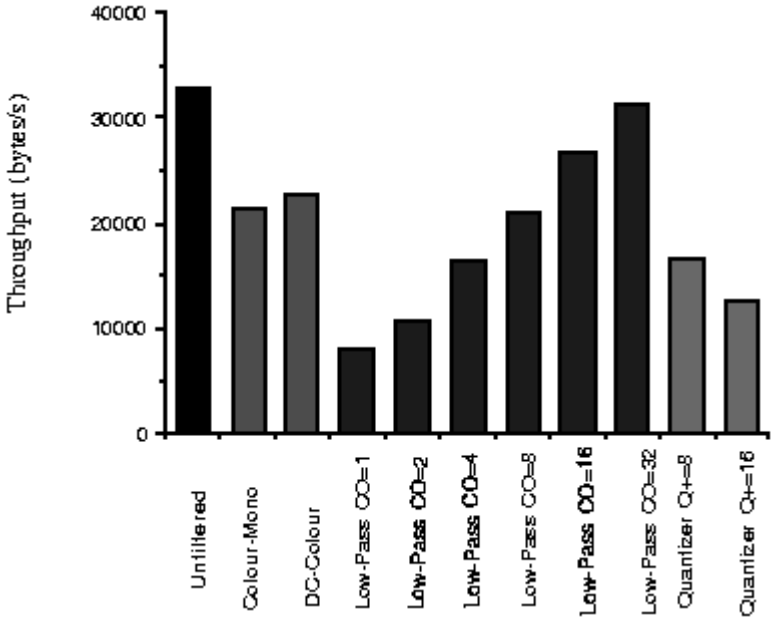


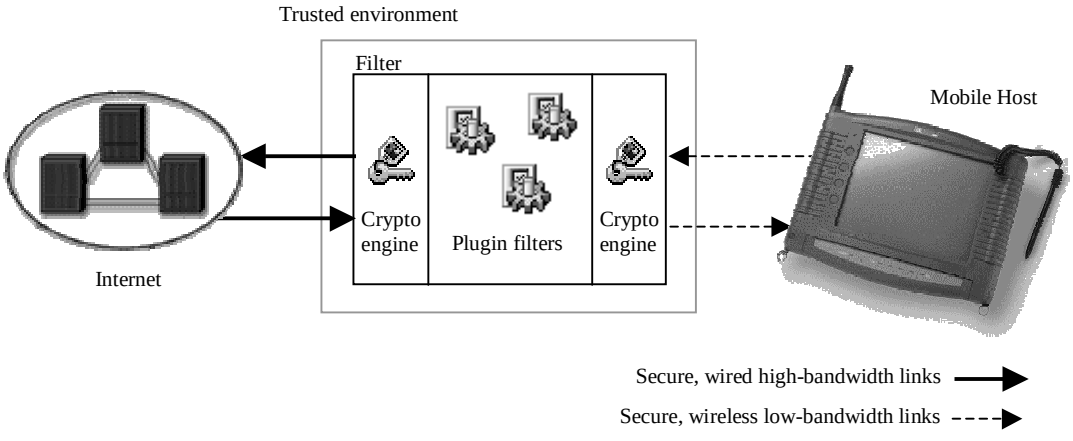
Figure 5.1: filter effect on throughput.
Source [Yeadon96] p.121

Yeadon developed a number of in-line adaptation filters that operate with a minimum of processing, involving some or no decompression. Frame dropping, where groups of packets are labelled as constituting one frame, is one example. This is particularly useful if one packet in a frame is missing; the filter knows it can drop the other packets of the then-corrupt frame. In-line translation requires an in-depth knowledge of encoding and compression schemes used.

The UCL Transcoding Gateway (UTG) also provides access to multicast conferences for hosts with only unicast connectivity. It is a heavyweight active services platform that will usually be run on a workstation by a user without local multicast support or adequate bandwidth to fully participate in sessions.

The UTG runs on a station with multicast facilities, and forwards conference data over a point-to-point link to the client. It similarly multicasts data that the client sends back. The UTG can also mix audio from a number of conference participants to form a single output stream, and transcode this data into a more bandwidth efficient format to send to its client(s).

Figure 5.2: filter operation
If the multicast session is encrypted, the media stream must be decrypted before filtering. The



link between the filtering machine and its client mobile host should also be encrypted.

The components in the UTG architecture perform the following tasks:

- The Real-Time Stream Protocol controller module provides the control interface to the unicast-only end-system.
- The access control module is used to verify that requests for transcoding and gatewaying are from authorised users.
- One or more media engines are instantiated to perform transcoding and gatewaying when required.
- Finally, the Mbus controller provides the necessary glue between all the other modules.

5.4 Congestion hints

Amir showed that SCUBA provides a scalable solution to sharing constrained bandwidth in multicast data delivery where a reasonable consensus among receivers can be reached and jointly acted upon by sources. But protocols that rely on a number of independent parties to act cooperatively can break down when one or more participants does not, particularly if participants can gain resources at others' expense by doing so [Savage99]. SCUBA must also be supplemented by filtering mechanisms like the UTG or layered coding schemes to address group heterogeneity. We have therefore developed congestion hints as an alternative mechanism that efficiently adapts multicast data flows to limited bandwidth paths they must travel over, without reducing the quality of data that well-connected hosts receive.

5.4.1 Source-based pruning

Version 3 of the Internet Group Management Protocol allows a user to receive data only from specified sources within a multicast group [Cain99]. Hosts multicast an IGMP message on their local network to join or leave groups, and can specify a list of up to 64 sources within that group to include or exclude from packets forwarded. The local multicast routers aggregate these messages and use the result with a multicast routing protocol to deliver matching packets and specify `(source,group)` flows they are interested in to neighbouring routers. This provides an elegant means by which the network can use group member preferences to reduce the volume of data sent via a multicast address.

5.4.2 Fractional source-based pruning

Amir describes two problems with using source-based pruning for lossy data. First, it forces an all-or-nothing decision upon receivers, when they may want to receive *some* traffic from a given source. Second, it does not help receivers who share a constrained path but are interested in different sources.

SCUBA allows all sources to intelligently share the available bandwidth – as long as receivers have similar patterns of interest. If not, they *all* may be dissatisfied with the resulting allocation. As an extreme case, if two recipients indicated 100% interest in two different sources, those sources would each use 50% of the available bandwidth, potentially leaving both receivers unhappy. While SCUBA can be used with media gateways to reduce the impact of divergent preferences combined with non-shared congested links, it does so only to the extent that such gateways are available throughout the network.

We have developed a fractional source-based pruning mechanism that overcomes these problems, allowing recipients to make their own decisions on sources of interest. Each recipient can provide hints to routers on the packets in each flow (defined by the source and destination) that they are more or less interested in receiving. When a router becomes congested, it can use these hints to drop packets on a more intelligent basis than schemes such as `droptail` [Floyd91]. This is a more generalized version of Bhattacharjee's active network architecture for MPEG support, which prioritises I-frames and ensures that groups of pictures are delivered entirely or not at all [Bhattacharjee97a]. Our protocol can also work

with unicast packets, to restrict congestion-control unaware flows and Denial of Service attempts.

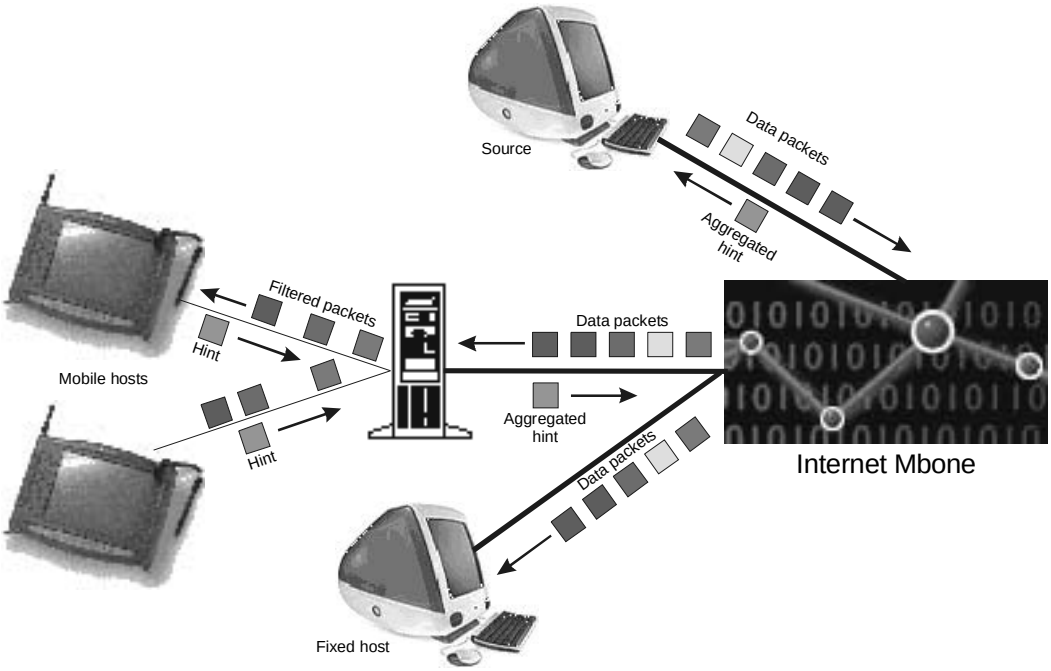


Figure 5.3: hint flow and effect

Hints operate in two directions. They can be sent by a multicast source to all members of the group, and specify the relative priority of packets – for example, Intra, Bidirectional and Predictive MPEG frames in a video stream. Receivers can also send hints upstream to a router specifying a filter to apply to incoming packets based on their source, destination and 16-bit IP identification number. Routers forward the hints back toward the source. Because hint packets have the Router Alert [Katz97] IP option set, each router will examine the packet. If it understands hints and is currently using them, it will add the hint to its filtering table. Otherwise, the packet is forwarded on with no further action.

5.4.3 Hint definition

The overriding priority for hints is that routers can act them upon very quickly. Therefore, there is only one hint for each `(outgoing interface, source, destination)` tuple. Whenever a hint arrives at a router it is written into the router's hint table, subject to the security checks detailed below. If a hint is already in place for a given multicast address, the two are combined so that the union of the two will be used to filter traffic on that interface.

Each hint contains a pattern that is matched against the bits set in each packet's sequence number when it is about to be written to a congested outgoing interface. Packets whose source and destination match a hint are only placed into an outgoing interface's queue if their sequence number does (for a forwarding hint) or does not (for a blocking hint) match the relevant hint pattern. Patterns are specified using a bit string of the same length as the IP ID.

Each hint contains the following information:

`sourceIPaddress:destinationIPaddress:pattern:action:timetolive`

The simplest hints instruct a router only to forward or drop a certain percentage of the packets from a given source in a session. A pattern of `0x1` matches against 50% of packets; `0x3` against 25%; and so on. This is useful for coding schemes such as intra-H.261 [McCanne96b] where each packet carries Application Data Units of equal priority. A timeout of zero is used as "remove hint" instruction.

A conference member might have a general policy of receiving 100% of packets from the current speaker in a conference, but only 25% from other members. A series of hints to implement this policy for 60 seconds, given these source and destinations and 8-bit sequence length, would be:

```
128.16.5.3:224.2.162.48:0x0:f:0
128.10.21.9:224.2.162.48:0x3:f:60
128.9.3.15:224.2.162.48:0x3:f:60
128.78.15.31:224.2.162.48:0x3:f:60
```

Each time the speaker changed (detected through an application-layer floor control protocol or user interface action) four new hints would be sent to change each source's pattern.

More complex patterns can be specified when packets have different priorities. In an MPEG stream, for example, dropping I-frames will cause a huge loss in quality. Sources can tell clients the relative priority of packets using an enhanced hint message multicast to all receivers:

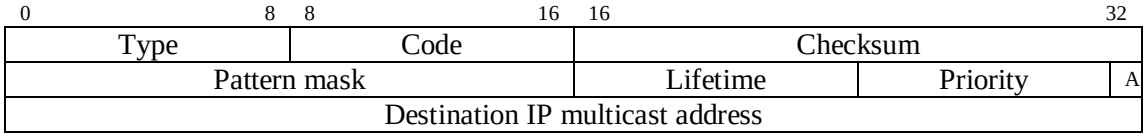
```
pattern:lifetime:priority:action:destination
```

`Pattern` specifies the pattern that all packets of this type will match: `action` specifies whether those packets should be forwarded or dropped. `Priority` is a 7-bit value whose mapping is decided by each source. One scheme for MPEG frames might be 128 for I-frames, 32 for P-frames and 4 for B-frames. Clients combine these hints with their own before sending them to a router.

Sources vary their sequence numbering to denote each different type of packet. An MPEG stream typically contains I, P and B frames in the ratio 1:2:9 [Hanzo94] depending on parameters such as degree of random access required. Packets large enough to contain an entire frame could therefore have the least significant bits of their ID header set as follows:

```
I: 0xE          P: 0x6          B: 0x1
```

Hints are transported in experimental Internet Control Message Protocol (ICMP) packets in the following format:



Type: 150 (experimental value)

Code: 0

Checksum: the 16-bit one's complement of the one's complement of the sum of all 16-bit words in the ICMP header, starting with the ICMP type.

Pattern mask: a 16-bit mask used to filter against IP IDs.

Lifetime: seconds until this hint expires.

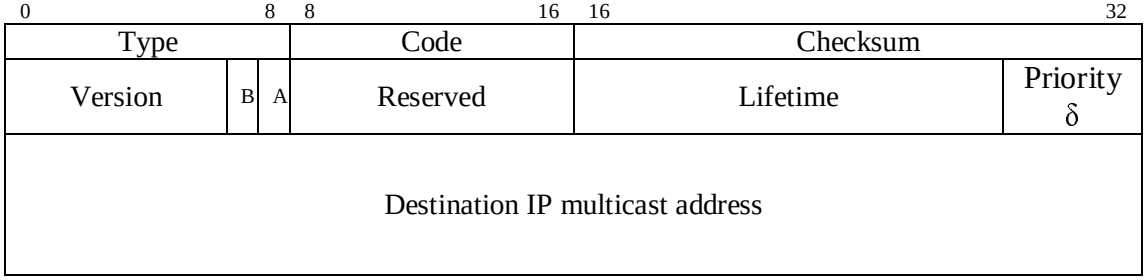
Priority: a 7-bit priority specified by a data source; must be ZERO from clients.

Action: should packets matching this hint be forwarded?

Destination IP multicast address: if the flow to be filtered is from a multicast group, its address is included here. Otherwise if null, the IP source and destination are used as the destination and source of the filter.

5.4.4 IPv6 hints

The original version of IP version 6 allows a simpler hint protocol. It discards the IPv4 identification header field but contains a 4-bit priority field we can use to adjust the priority of packets specified by the recipient. The updated IPv6 uses these priority bits for DiffServ, but the same effect could be achieved using a Hop-By-Hop option.



Type: 150 (experimental value)

Code: 0

Checksum: the 16-bit one's complement of the one's complement of the sum of all 16-bit words in the ICMP header, starting with the ICMP type.

Version: IPv6 hint protocol version.

Block: should packets matching this hint be discarded?

Action: is priority delta positive?

Reserved: reserved for later use, and to preserve word alignment.

Lifetime: seconds until this hint expires.

Priority delta: a 4-bit priority delta to be applied to packets matching this hint; positive or negative based upon the action bit.

Destination IP multicast address: if the flow to be filtered is from a multicast group, its address is included here. Otherwise if null, the IP source and destination are used as the destination and source of the filter.

Denial of service traffic can therefore be filtered using the block bit, while the priority of packets from different sources in a multicast group can be increased or decreased according to the recipient's interest.

5.4.5 Hint aggregation

Yeadon demonstrated that a tree of filters could be formed from a high-quality source that gradually lowers the video quality to the level required by each branch point or node in the network [Yeadon96]. This makes maximum efficient use of bandwidth: there is little reason to send a high bandwidth stream over a potentially congested link if it will simply be filtered at a later point in the network.

Congestion hints can be aggregated in the same way. This is also important to prevent a "hint implosion" where hint messages from many receivers overwhelm the source and link capacity. When routers receive hints from downstream nodes they update their filtering table, calculate the union of those filters for the affected flow and send one aggregate hint upstream. A timeout is then set before which no new filters will be sent.

All hints are soft state: they include a timeout value after which they will be deleted by routers. Sources and receivers must periodically refresh this router state by re-sending hints. Nodes also resend hints when a Source Path Message shows that the route back to the source has changed. This functionality is optional for routers, which can reduce processing by

disabling it at the cost of receiving unfiltered traffic until the usual state refresh processes cause a new hint to be sent upstream.

These procedures ensure that hints are scalable and robust, like other soft-state protocols. Each node sees at most one hint message per timeout period per child, scaling workload O(log *receivers*)/timeout. Any loss of state in routers will be fixed within one timeout period when downstream nodes send updated hints. The same simple filtering algorithm is used at all points in the network and can even run at sources, allowing output to be scaled back at that point if possible – allowing network stacks to scale back the output of congestion-control-unaware applications. Hint-compliant applications can use the information provided as input to their congestion-control mechanisms. Hints therefore act as a distributed superset of the functionality of SCUBA.

5.4.6 Service mobility

The last-hop wireless link to a mobile host usually represents the most congested link in an IP route. Congestion hints are therefore particularly useful in wireless base stations, even if implemented nowhere else in the network.

By definition, mobile hosts rarely stay in the same place for much time. If they are moving small distances close to their user's "home," it isn't too inefficient for their traffic to be routed through their home base. Mobile IP [Perkins96] routes traffic to a mobile host via their "home agent" which keeps track of their location. This data goes to a "foreign agent" on the MH's local network, which finally forwards it to the MH. In this scenario, the home agent is an ideal place to run active service code [Zenel97]. The trusted workstation running that code is the security environment assumed by many active service schemes.

Unfortunately, this also causes high latency if the MH is some distance from home, which is particularly damaging for real-time data. Mobile IP extensions [Perkins97] allow the sender of traffic to an MH to discover the location of its foreign agent from its home agent, then from that point forward communicate directly with the foreign agent. But this means that the base station is often the only point through which all traffic to an MH is routed. This is also the case for multicast data when the MH has chosen to subscribe its local foreign agent directly to a group rather than have its home agent forward the group traffic. Active services must therefore be run at each base station the MH moves between, causing potentially high handoff delay if the service is not already running at that point.

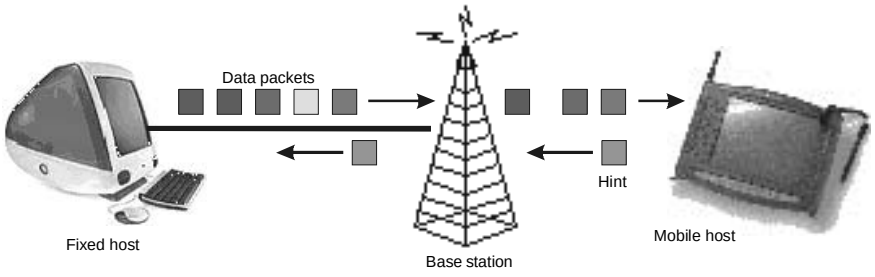


Figure 5.4: base station using and forwarding hints

Because congestion hints are a lightweight mechanism, a mobile host simply needs to send a hint to a new base station as it arrives, or even before. Hints could easily be integrated into Berkeley's fast handoff mechanism, where an MH primes a set of secondary base stations within range to be ready to quickly start providing services if the MH hands off to one of them [Balakrishnan95]. Hints sent to secondary base stations should have a Time-To-Live value of 1, so that the hint does not propagate further into the network. When the MH switches to a new primary base station it should send another normal hint that the base station will forward back through the multicast tree to its source, setting up filtering on this new route.

5.4.7 Flow fairness

Most current congestion control mechanisms require the co-operation of users. TCP clients, for example, start new connections at a low rate, increase that sending rate slowly, and halve the rate when they detect congestion. “Rogue” implementations could ignore these restrictions and gain bandwidth at the expense of other co-operative users [Savage99], but the kernel programming required to create such implementations is not trivial [Henderson01].

Hints have the protection that their use may provide a better quality of service to cooperative individuals. Studies have found that predictable quality increases users’ perception of multimedia quality above that of a less stable higher bandwidth connection [Bouch00]. Individuals therefore have a selfish personal motive to use hints.

Greedy users may still attempt to take advantage of co-operative users sharing the same links. By ignoring congestion control mechanisms, they may be able to drive their bandwidth share higher at the expense of flows being filtered by hint-compatible routers.

The only long-term solution to uncooperative users in any congestion control scheme is to enforce fairness within the network. Proportional charging of users or their service providers for their use of the network is an economically efficient way to do this. Rather than paying a usage charge or for specific quality of service, users pay their service providers a small fee to cover fixed costs such as buildings and personnel and then are only charged when their traffic causes congestion in the ISP’s network. This produces the socially optimal use of the network: the marginal cost of uncongested bandwidth is virtually zero, so users are encouraged to make full use of the resource but not push their bandwidth consumption beyond that point [Henderson01].

Edge pricing using Explicit Congestion Notification has been suggested as an efficient mechanism to implement charging. Routers that are experiencing congestion mark ECN bits in packet headers; users that receive these packets are then charged in some form by their ISP [Henderson01]. Uncooperative users are therefore charged for creating congestion in links.

ECN could even be used to encourage the use of hints: routers would mark proportionately fewer packets in a flow where hints were present.

5.4.8 Authentication

Hints must be strongly authenticated to prevent trivial Denial of Service attacks. However, we cannot use cryptographic authentication in all cases as this would be too resource-intensive for routers and would provide a new mechanism for further denial of service attempts. Verifying a public-key signature is an expensive operation, and currently unsupported by IPSEC. A client cannot set up a security association with every router it wishes to send a hint to in order to allow symmetric authentication. Routers also aggregate hints as they travel up a multicast tree, which breaks source-based authentication.

Therefore we need to use the protective infrastructure methods used by network operators [Blake98, Brown01]. Ingress networks must check that the source address contained in a hint matches the address currently assigned to the originating host on a given link. If that link is connected to a shared subnet or its physical security is in doubt, cryptographic authentication may be required on that first hop. Peering networks that support hints should allow them to flow unchanged between the two domains. But when a hint-compliant network receives a hint packet from a network that does not support the protocol, it must drop the packet.

Hints are only used by routers when examining packets travelling onto the interface that the hint arrived on. Routers therefore drop any incoming hint that arrives on an interface different to where a packet would be routed for its source, as it is redundant. This means that spoofed hint packets arriving from a different source that have managed to bypass the network’s security mechanisms will only have an effect if they arrive on the correct interface. As hints have a maximum lifetime of only 256 seconds, the impact of successfully spoofed hints is also bounded to that period.

Network operators may wish to implement further protective measures within their networks, such as cryptographic authentication of specific links.

5.4.9 Packet filter performance

To compare the performance of packet filtering against that of an application-layer media gateway, we compared a filter implementation in the IP stack of FreeBSD 3.4 against the UCL Transcoding Gateway. Kernel-level code integrated into the queue management processes implement packets filters, while a user-level process sets them up. The UTG runs as a user-level set of processes.

We ran tests using system shown in figure 5.5. A Solaris workstation was connected over a 100Mb/s switched Ethernet to an AMD K6 233MHz host running FreeBSD 3.4. This forwarded data to another Solaris workstation. A route was set up on the first host to send data to the second Solaris workstation via the intermediate host. System load was recorded on the intermediate host using the BSD `top` tool.

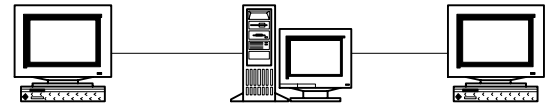


Figure 5.5: testbed configuration

Five minutes of footage from a fixed webcam was sent using `vic` from the first host to the intermediate host, which carried out one of the following types of filtering before sending the data on to the end-point workstation:

1. Forward: a UTG forwarded a variable percentage of an H.261 video stream from the base station.

2. Transcode: a UTG transcoded a JPEG stream to an H.261 stream at the base station.

3. Filter: packet filters were used on an H.261 stream at the base station.

The webcam was focussed on a poster board. Random light variations in the room combined with a low-quality webcam caused the image to be continuously updated by `vic`.

5.4.10 Results

Resource consumption: we first measured the CPU and memory usage of the UTG and packet filters using a Unix process monitoring tool; the averages are shown in figures 5.6 – 5.8. The filters used such negligible amounts of CPU time and memory that they could not be accurately measured, as shown by figure 5.9. The UTG used 1.27% and 0.85% CPU time and 3494Kb and 4072Kb memory respectively. The UTG is particularly unscalable as it runs a new copy of itself for every stream being processed.

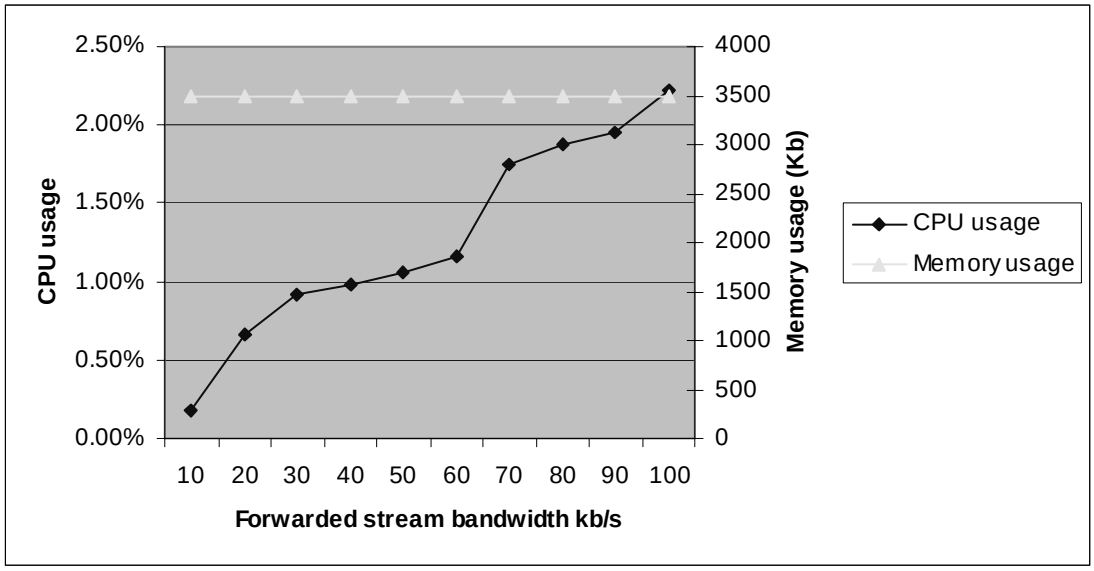


Figure 5.6: UTG selective forwarding memory and CPU usage

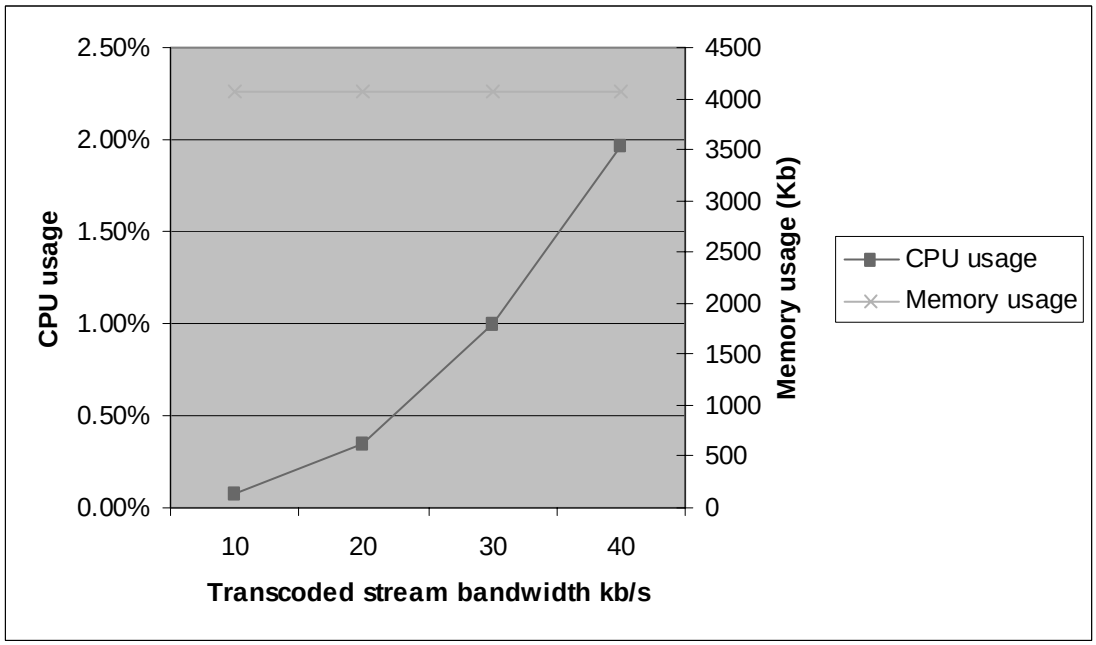


Figure 5.7: UTG transcoding memory and CPU usage

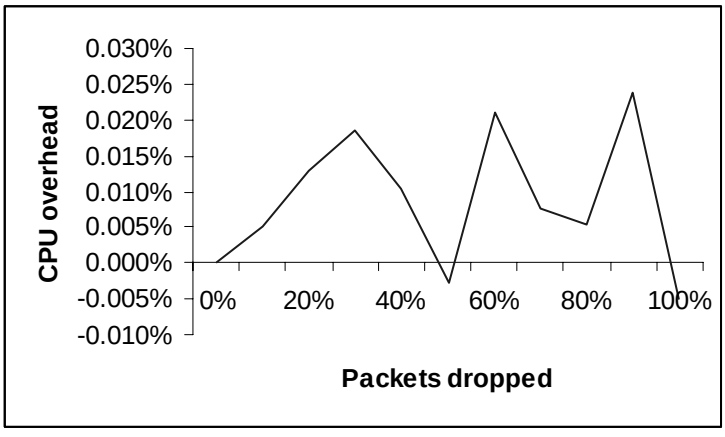


Figure 5.8: Packet filter CPU usage

We also examined the overhead incurred by the cryptographic processing required at the UTG to decrypt then re-encrypt DES-encrypted streams. For a 100Kb/s stream, this added 2.37% CPU overhead.

Bandwidth utilisation: we then analysed how effectively filtering at different levels of granularity uses available bandwidth.

The simplest scenario to analyse is a uniform random distribution of available bandwidth *b* and a layered codec that divides that bandwidth into *l* equally sized layers. Bandwidth utilisation as a proportion of the total available is then:

$$\frac{2\sum_{i=0}^{l-1}\left(\frac{b}{l}\right)^2 i}{b(b+1)} \equiv \frac{b(l-1)}{l(b+1)}$$

(5.1)

Figure 5.9 shows the shape of this function for 0-10 layers over the bandwidth range 0-1000kb/s:

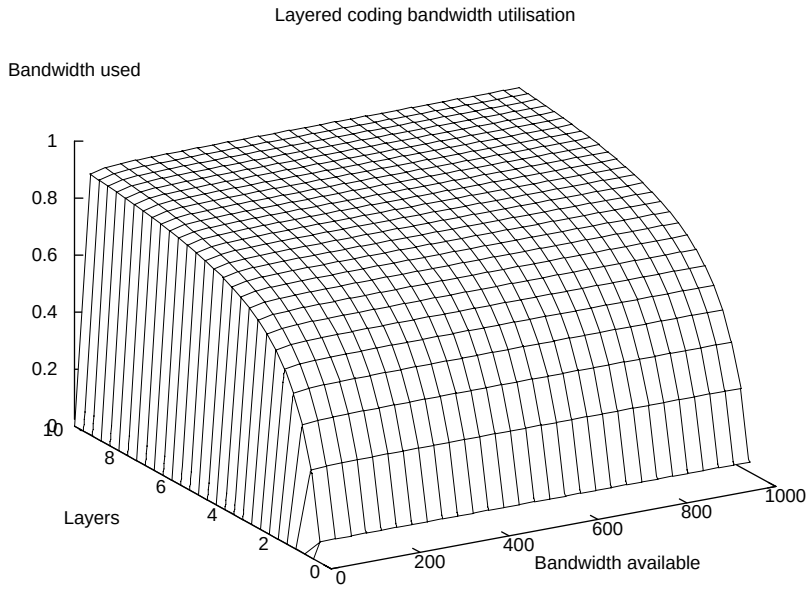


Figure 5.9: bandwidth utilisation in equal-layer schemes

Where McCanne's exponential layering scheme is used, the layers are sized 32 * 2^m kb/s, m=0…*l* [McCanne97]. This leads to an ideal bandwidth utilisation of:

$$\frac{2^{11}\sum_{i=0}^{l-1} 2^i(2^i-1)}{b(b+1)} \equiv \frac{2^{11}}{b(b+1)}\left(1+\frac{4^l-1}{3}-2^l\right)$$

(5.2)

This function is shown in figure 5.10 for bandwidth usage between 0 and 1000kb/s.

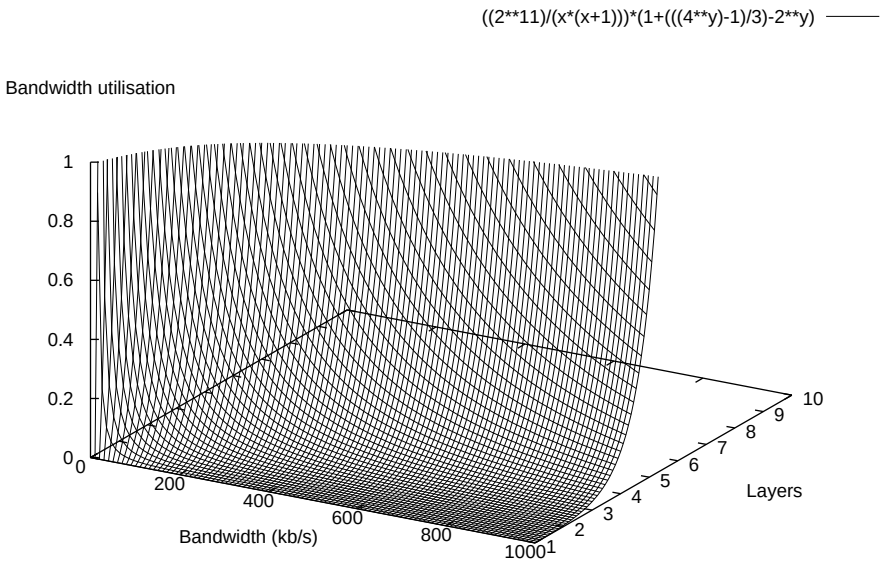


Figure 5.10: bandwidth utilisation in exponential-layers schemes

The actual bandwidth utilisation of layered multicast schemes is not ideal due to several factors. The codecs used can generally only produce fixed layer sizes. There is a time lag between changes in network conditions, their detection by clients and finally their notification to servers. And any given set of multicast recipients will have continuously differing amounts of bandwidth available. An end-to-end scheme that attempted to deal with all of these variations would most likely make the problem worse with large oscillations in output caused by delayed feedback and significant bandwidth consumed by control traffic.

User perception of quality: we finally examined whether transcoding produced perceptually superior video quality to packet filtering. Using the experimental system shown in figure 5.4 we sent a video stream via an intermediate host to the recipient host, where it was recorded using the `rtpdump` tool. The video stream was a 30 second clip from the action movie *Enemy of the State*. In our first condition, the clip was in JPEG format, and was transcoded by a UTG running at the intermediate host into H.261 format. In our second condition, the same clip in H.261 format was filtered at the intermediate host using BSD's packet filter to produce another H.261 stream of the same size as the transcoded stream.

We played the two resulting recorded clips in random order to 12 participants using `rtpplay` and `vic`. After each clip the participant was asked to rate the quality of the clip using the unlabelled quality scale developed by Watson [Watson98] and shown in figure 5.11, with the top of the scale orally described as "the best quality you can imagine" and the bottom as "the worst quality you can imagine". After both clips were played, participants were asked which they considered to be of better quality.

Figure 5.11: quality scale

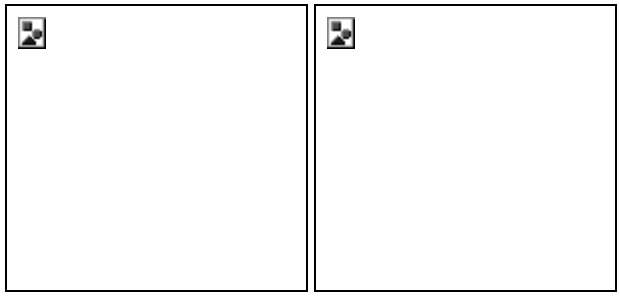


Figure 5.12: filtered and transcoded clip playback

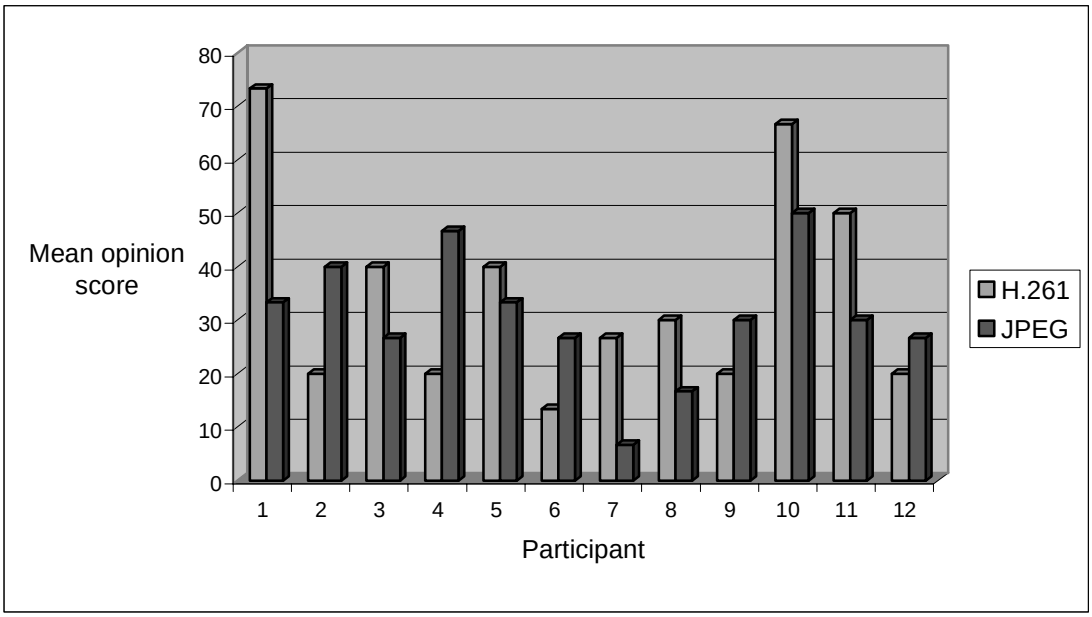


Figure 5.13: quality ratings of transcoded and filtered video

The mean and standard deviation of the 100-point quality ratings were 30.56 and 11.88 for the transcoded stream and 35 and 19.57 for the filtered stream; the distribution is shown in figure 5.13. A t-test shows that there was no significant difference between these ratings (p=0.451251, two-tailed). We can therefore conclude that for such low bandwidth streams, transcoding did not produce a perceptually better result for users.

5.5 Distributed defence against flooding attacks

Distributed denial of service attacks were first seen during the summer of 1999. An attacker exploits security holes in popular software to install "zombie" processes on hundreds or thousands of Internet-connected machines around the world using automated tools. He then issues a single command to each of these machines that causes them to start sending high volumes of traffic toward a target machine. The total volume of this traffic causes the target or its network to crash. In February 2000 many major e-commerce sites such as Amazon, Yahoo and E*Trade were hit by these attacks [Todd00]. The Code Red worm seen in summer 2001 performs the initial zombie installation automatically; each infected machine seeks out others to attack, before trying to flood a specific IP address (at the White House) on a certain date [CERT01]. Over 359,000 machines were reported to be infected at the height of the epidemic [Moore01].

The end-to-end solution to these attacks would be to ensure that all routers and hosts in the Internet could at least drop packets at the full line speed of inward links. This reduces the impact of attacks as machines do not crash, but legitimate traffic to and from sites can still be blocked.

Hints can be used in a distributed defence against the effectiveness of such attacks. When an attack is detected, the target machine or network can send hints upstream blocking all packets from the zombie machines that are sending the flood traffic. This can also serve as an automated alert to ISPs on that route that their networks are being used as part of an attack, allowing them to take more sophisticated countermeasures. The availability of links for legitimate traffic is therefore maintained.

This approach will be most effective once egress filtering is widely implemented by networks connected to the Internet, preventing zombies from using random addresses in the packets they send. Otherwise attack flows will not be identifiable using one source address. Targets should only send hints once they have received a reasonable number of attack packets from

one given source address, and should rate limit the total number of hints they send to a small percentage of the bandwidth available to prevent the hint packets making the problem worse.

5.5.1 Performance

The FreeBSD operating system communications stack contains an efficient packet filtering module. We ran tests on a typical desktop machine (an AMD K6 233MHz host, shown in figure 5.4) running FreeBSD 3.4 to examine the ability of such a system to protect a small subnetwork against a distributed denial of service attack.

The filtering machine was connected over a 100Mb/s switched Ethernet to twelve other similar hosts running a mixture of Solaris and FreeBSD. A route was set up on one host to send a one-minute video stream to another workstation via the filtering machine. Meanwhile, we ran another program on one to ten other hosts that simulated a flood attack by sending UDP packets to the filter machine at a user-configurable rate and size. We measured the load on the filtering machine using `top` and the percentage of the video stream that reached its recipient using `tcpdump` under two conditions: using the 400 byte average packet size found at midday on two large Internet backbones [Thompson97] and the smallest possible IP packet of 20 bytes. The latter are easily spotted by intrusion detection software as packet flooding attacks whereas the former are not given away by their size. Figures 5.14 and 5.16 show the system load caused by filtering; figures 5.15 and 5.17 show the effect of the attack on the forwarding of one 'good' video stream.

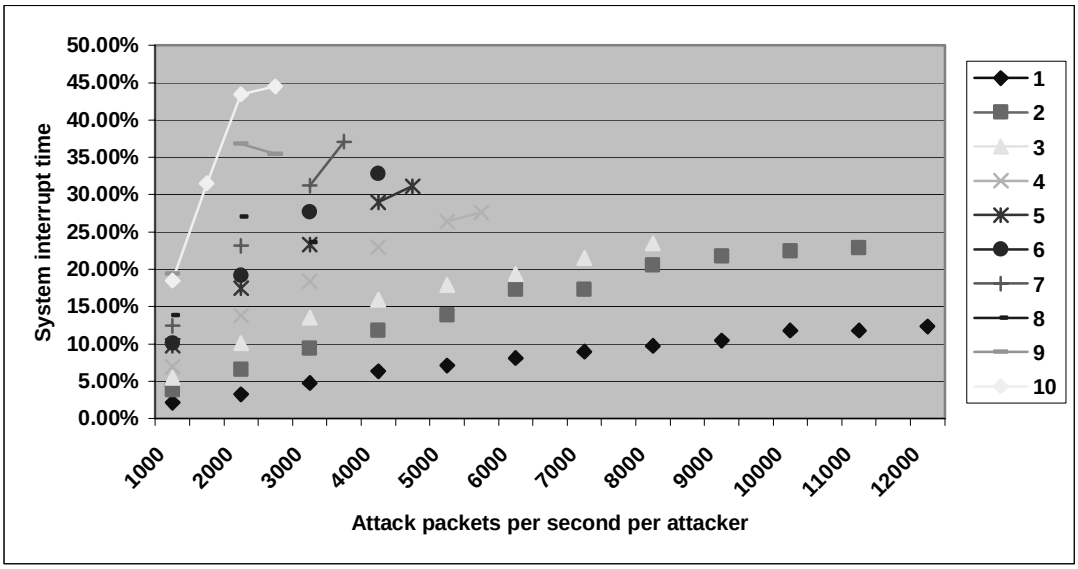


Figure 5.14: system load on filtering machine by number of attackers (400B packets)

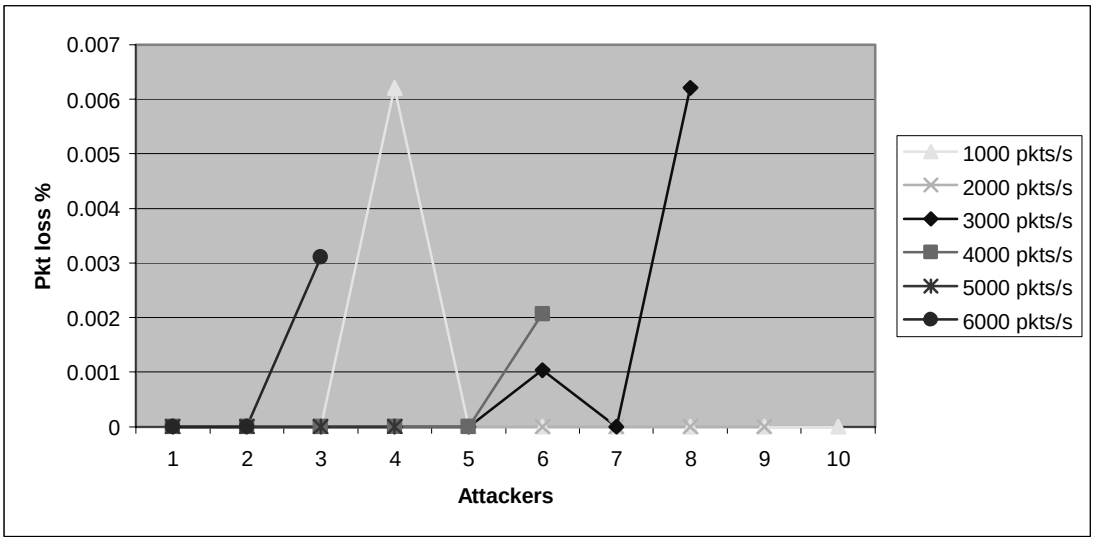


Figure 5.15: effect of filtering DDoS traffic on one "good" stream (400B packets)

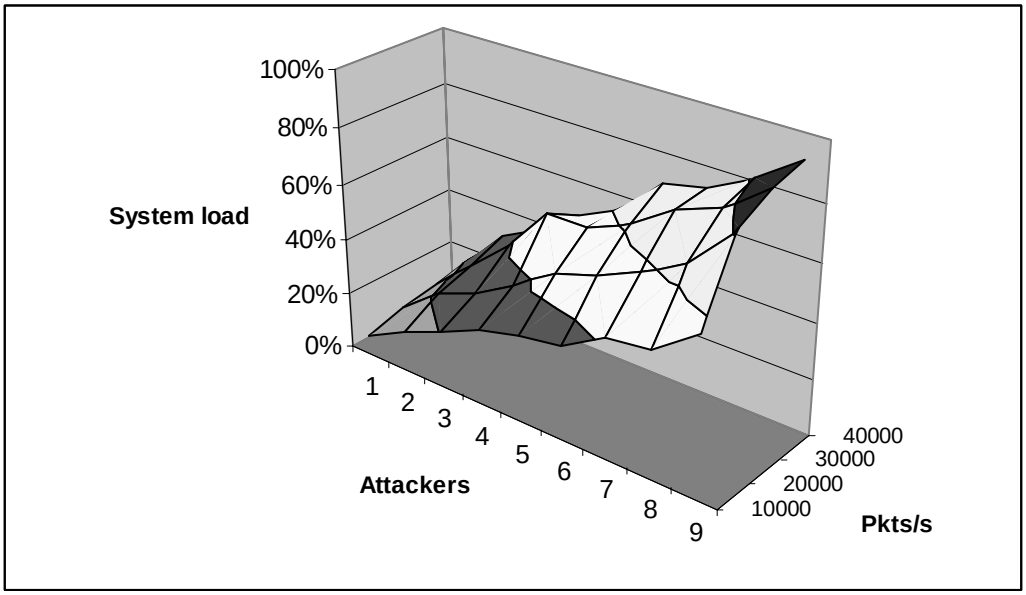


Figure 5.16: system load on filtering machine by number of attackers (20B packets)

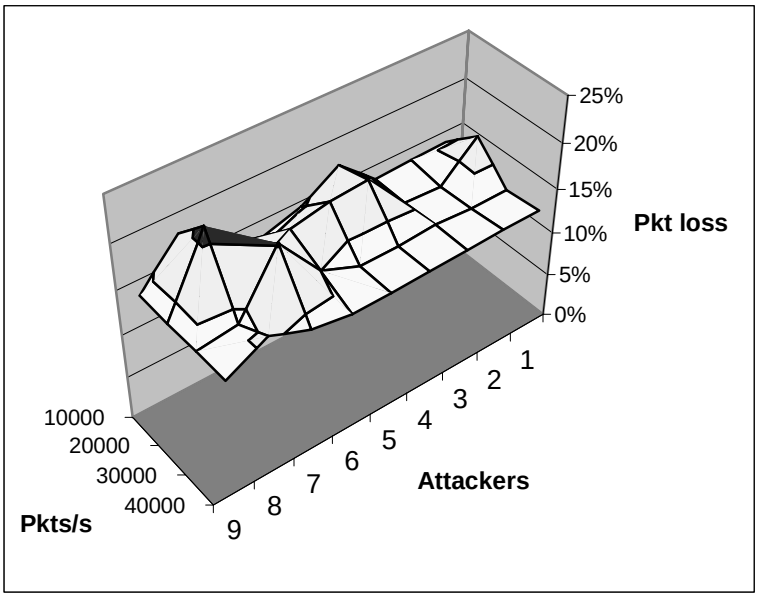


Figure 5.17: effect of filtering DDoS traffic on one "good" stream (20B packets)

Inexpensive personal computers can therefore be used to protect local subnetworks against internal and external denial of service attacks. They can also pass filtering requests upstream to the higher-end routers within local networks and their Internet service provider, where hardware assistance allows much high filtering rates. The Cisco 6500 router can filter out around 2.4Gb/s of attacks using 20b packet floods. It can restrict traffic from other common attacks even more effectively. Smurf attacks broadcast spoofed ICMP echo requests using the address of the target as the source address: all recipient machines then return 28-byte minimum reply packets, which the Cisco 6500 can filter at 3.36Gb/s. The TCP-SYN attack sends 40B minimum TCP setup requests, which can be filtered by the Cisco at 4.8Gb/s.

Figure 5.18: UCL network and filtering capacity

Figure 5.19: Department of Computer Science network

Figures 5.18 and 5.19 show how UCL and its Department of Computer Science's networks can be protected using this scheme. UCL's backbone network is configured as two redundant parallel stars, connected using Cisco 6500 routers. The gateway router to the London Metropolitan Area Network can individually filter 1/4 of its total capacity of 10Gb/s against packet floods. The two internal core routers can isolate up to two compromised internal building networks running at the full link capacity of 1Gb/s. And each building network can protect each of its departmental links from the others, currently running at 10Mb/s or 100Mb/s. The computer science core network connects to the Pearson building router at 10Mb/s, which its central Cisco 4700 router can filter as well as isolating the four subnets connecting at 10Mb/s.

It is important that sites do not rely solely on filters at a small number of external gateways: internal machines may become infected and start attacking other local hosts. But protecting against outside attacks is most important as internal systems are under local administrative control and can be shut down relatively quickly as a temporary protection. UCL's three-hop network layout would allow almost total protection against denial of service attacks if configured with hints.

5.6 Conclusion

In this chapter, we have described several different mechanisms that can be used to shape multimedia flows to available bandwidth. End-to-end schemes such as layered multicast allow clients to receive a stream at varying levels of quality, but can only respond slowly to changes in bandwidth availability. Media gateways can filter and transcode streams between various formats, but require plaintext access to do so.

We therefore developed a lightweight distributed packet filtering mechanism that efficiently adjusts the bandwidth used by appropriately coded streams, and can also act as a defence against denial of service attacks. We showed that it makes more efficient use of bandwidth than layered coding schemes, and that its performance requirements are far lower than those of the UCL Transcoding Gateway despite producing comparable end-user perception of the quality of filtered streams. We also demonstrated that it could effectively protect small networks against denial of service attacks, and showed how it could be combined with hardware router filters to protect large networks.

The main research response to DDoS attacks has been the development of schemes that allow packets with spoofed source addresses to be traced back to their origin. The IETF ICMP Traceback working group is developing a protocol whereby routers send ICMP iTrace packets containing route information towards the destination of around 0.05% of packets that they forward. When a packet flood is underway, this will result in enough traceback messages reaching the destination to allow its true source to be determined [Bellovin00]. "Intention-driven" traceback allows a recipient network to signal whether it is interested in receiving iTrace packets, increasing the proportion of messages that will be useful to the receiving network [Mankin01]. Other schemes concentrate on marking a percentage of packets with partial route information [Savage00, Song01], or record-keeping by routers of forwarded packets using efficient representations such as Bloom filters, allowing recipients to query routers to determine the source of a given packet within a limited timeframe [Snoeren01].

These schemes are important, particularly while ingress filtering is not universally implemented, but do not allow the immediate protection of networks under attack. A combination of filtering and traceback therefore provides good "defence in depth."

A hint-like mechanism is particularly important in networks that implement any kind of 'receiver pays' charging. If hosts have no way of rejecting unsolicited traffic, denial of service attacks will be very costly to recipients. And by induction, networks that have any bandwidth-based charges on interconnects want to stop such traffic before it crosses a peering point. A hint-enabled network will therefore have a large fiscal incentive to insist that peering networks also act upon hints.

These bilateral agreements also better reflect the trust relationships between networks, and the potential costs of attacks. Rather than requiring all networks to trust all other networks (in order to allow flows to be blocked at source once egress filtering is universal and a protocol to do so had been designed) networks can build upon their pre-existing relationships to reduce traffic flows that may incur mutual costs.

6 Increasing TCP performance over lossy links

"Slow and steady wins the race."

–Aesop

6.1 Introduction

The Transmission Control Protocol is used to provide a reliable stream abstraction over the best-effort Internet Protocol [Postel81]. It contains a number of features that ensure reliable, in-order data delivery to applications, and prevents limited bandwidth links from being overloaded with traffic using congestion control mechanisms.

The increasing use of wireless links has caused problems for these mechanisms. TCP interprets segment loss as congestion and reduces the speed at which it is sending data, because the wired Internet on which it was developed is largely reliable. But higher data loss on wireless links causes TCP throughput to be a small fraction of bandwidth available, because loss causes an exponential scaling back of transmission that then only increases linearly as segments are successfully received.

Active network solutions have been proposed that cache and retransmit segments across lossy links and correspondingly alter the acknowledgements travelling back to the sender. But these will not work if IPSEC is in use, as the acknowledgements are cryptographically protected.

In this chapter, we describe these systems and an alternative we have developed that works with end-to-end network-layer security by building on link-layer reliability whilst preventing inefficient conflict with TCP's reliability mechanisms. We further show how fast handoff can be supported, and how our scheme can be used to enhance the performance of reliable multicast protocols.

6.2 Reliable unicast

TCP provides a reliable, in-order data delivery service to applications. It sends information in segments using IP packets that may be lost or delivered out of order. The recipient orders and acknowledges data received; the sender re-sends segments if an acknowledgement has not been received within an adaptive timeout or three later segments have been acknowledged [Braden89]. This process is entirely transparent to applications, which see only a stream of data.

6.3 Congestion control

IP provides little feedback to senders on the rate at which they can transmit packets without causing congestion in the network. Mechanisms such as source quench designed to provide such feedback have so far been unsuccessful due to their increased network and gateway load and difference in implementation. A source quench message could mean that a gateway is experiencing or expecting congestion, suffering from a short or heavy burst load, or other variants depending on router manufacturer [Mankin91].

TCP originally limited a source's sending rate only to the amount that could be accepted by the receiver, but this led to unstable behaviour once load became even moderately high or two hosts on fast local networks attempted to communicate over a slower intermediate link [Jacobson88].

Newer TCP implementations therefore "probe" the network capacity by sending information over a new connection slowly, then increasing and decreasing this rate according to segment loss, using the following algorithm [Allman99]:

1. When a new connection is opened, set `transmission_window` to that advertised by the receiver, `threshold` to 64K and `congestion_window` to `sender_max_segment_size`.

2. Send as many segments as will fit in `min(transmission_window, congestion_window)`. While `congestion_window < threshold`, increase `congestion_window` by `sender_max_segment_size` for each acknowledgement received. While `congestion_window > threshold`, do the same for each transmission window's worth of data acknowledged. Update `round_trip_time` to `0.9 * round_trip_time + 0.1 * latest_trip_time` for each acknowledgment. Set a timer counting down from `round_trip_time + 4 * δ(round_trip_time)` for each segment.
3. If a segment has not been acknowledged before this timer reaches 0, set `threshold` to `congestion_window/2`, `congestion_window` to `sender_max_segment_size` and double `round_trip_time`.

Fast retransmit is an extra mechanism that reduces the time taken to retransmit a lost segment. When a source receives three duplicate acknowledgements for a given segment, they assume the next segment sent has been lost rather than just reordered in the network. The segment is therefore retransmitted immediately, rather than when its timer expires. The first fast retransmit implementations then followed the congestion control procedure triggered by a segment timer expiring (step 3 above). But since three duplicate acknowledgements indicates that segments (at least three) are still arriving at the receiver, newer implementations go into fast recovery mode rather than slow start. The source sets `threshold` to `max (segments_in_flight_size/2, sender_max_segment_size)`, retransmits the lost segment, then sets `congestion_window` to `threshold + 3 * sender_max_segment_size`. For each further duplicate acknowledgement that arrives, `congestion_window` is increased by `sender_max_segment_size`. Once new data is acknowledged, `congestion_window` is deflated to `threshold`.

6.4 Reducing the impact of lossy links

The high error loss rate of wireless links can severely reduce the performance of TCP connections running over them. Because TCP stacks continue to interpret such loss as congestion, they invoke congestion control mechanisms, reducing the rate they send data. This is a sensible course of action over a wired link, but incorrect when a wireless link is involved. Higher loss rates on the wireless hop would best be counteracted by the opposite behaviour, *increasing* the send rate.

6.4.1 Indirect TCP

The first proposals to address this problem split a TCP connection into two at the base station of the wireless link. With Indirect-TCP, TCP is used between the sender and base station, and a protocol optimised for the wireless link used between the base station and a mobile host [Bakre94]. The two connections can deal better with the different link properties of the wired and wireless sides. Unfortunately this breaks the fundamental property of TCP: its reliability. Received segments can be acknowledged by the base station, causing the sender to believe they have been safely delivered, but then be lost before arriving at their destination. This is disastrous for any protocol dependent on TCP for reliability. It also requires applications on the mobile host to be relinked with a special communications library, and careful implementation to avoid the protocol gateway imposing a heavy performance penalty [Balakrishnan95].

6.4.2 TCP snoop

To gain the benefits of I-TCP without these drawbacks, Balakrishnan suggested modifying the end-to-end behaviour of a TCP link at the base station where traffic moved between the wired and wireless domains. Code running at that point caches IP packets until the wireless recipient acknowledges the TCP segments they contain. When the base station sees duplicate acknowledgements from the wireless host, signifying lost segments, it retransmits those

segments with a high priority. The acknowledgements are modified to prevent the wired host knowing about the loss. The base station also monitors TCP data travelling in the other direction, and sends negative acknowledgements (NACKs) to the wireless host for segments detected missing. The wireless host then retransmits those segments with high priority. This reduces the time taken to signal segment loss from the round-trip time between the communicating wireless and fixed hosts to the RTT between the wireless host and base station. Balakrishnan found that these two processes, which he named `tcp-snoop`, increased the performance of TCP connections over wireless links by up to 20 times [Balakrishnan95].

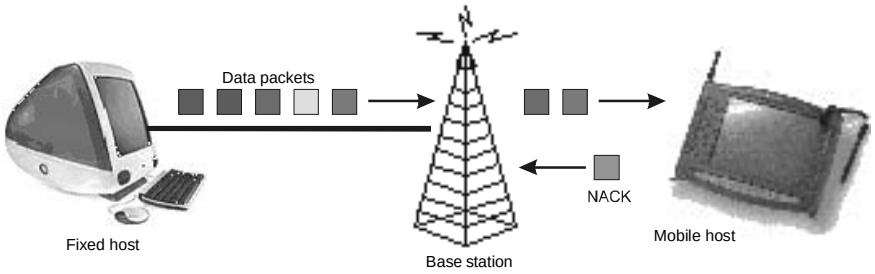


Figure 6.1: base station manages retransmissions and NACKs

6.4.3 Working with network-layer security

Unfortunately, IPSEC flows cannot be processed by active services such as `tcp-snoop`. Because the transport-layer TCP acknowledgements are protected, they cannot be read if encryption is used, or altered if authentication or encryption is used, by intermediate points.

Bellovin suggested the discovery header as a solution to this and related problems [Bellovin99]. This is a plaintext option header in an IP packet that contains copies of the important information encrypted within a packet's Encrypted Security Payload. While this would allow `tcp-snoop` to see TCP sequence numbers, enabling caching and detection of lost segments, the snoop code would be unable to alter the acknowledgements contained within the ESP. The remote sender would therefore still reduce its sending rate upon seeing segment loss over the wireless link. Balakrishnan found that acknowledgement suppression led to an increase of 30% throughput during an 8MB transfer over a Wide Area Network. Without this suppression, 90% of segments were retransmitted by both the source and the base station [Balakrishnan97].

Without splitting the security into two parts joined at the base station, or providing the base station with the secret cryptographic keys needed to alter information in packets, there does not seem to be a way to allow `tcp-snoop` to work with IPSEC flows. Both of these 'solutions' have the same effect on the connection's security as I-TCP does on reliability. They provide an attacker with a new point at which data can be intercepted and altered without alerting the sender or receiver.

6.4.4 Using link-layer reliability

Many link-layer protocols such as GSM (Global System for Mobile communications) have options for reliable transmission. GSM uses a combination of Forward Error Correction (FEC) data and retransmission of lost or damaged blocks to ensure data is transported reliability across a wireless link. Blocks are interleaved over a number of timeslots to further reduce the impact of block erasure [Rahnema93].

Because link-layer protocols run beneath IPSEC, they have no effect on its operation. The link layer neither knows nor cares what data its frames are carrying, plaintext or ciphertext. Link-layer reliability therefore does not have the same negative effects on end-to-end security as network or higher-layer schemes.

There is a danger that running a higher-layer reliable protocol like TCP over a reliable link-layer protocol will lead to damaging interaction between the error-correcting facilities of the two. For example, Kojo found that if GSM tried to retransmit damaged data several times, the latency could cause the TCP sender to time out and retransmit that data, unnecessarily overloading the link [Kojo97]. However, Ludwig found that this was avoided if the TCP implementation more accurately measured the connection's round-trip time, adjusting quickly to changes in delay [Ludwig99].

Higher-bandwidth links with reliable but out-of-order delivery of frames can interact particularly badly with TCP. By the time the reliability protocol has retransmitted a lost frame, later frames have often already caused the receiving TCP stack to send enough duplicate acknowledgements to invoke the sender's fast retransmission and congestion control mechanisms [Balakrishnan97].

Chaskar et al. showed that reliable, *in-order* delivery of data is key to successful TCP operation. Their analysis and simulations demonstrated that link layers providing this property allow TCP throughput to average 75% of the effective bandwidth of the bottleneck wireless link. This is the standard behaviour of TCP in its congestion avoidance phase across **any** link [Chaskar99].

The only requirement for this result is that the base station adequately buffers data so the probability of overflow at that point is lower than $1/W_{bd}^2$, where W_{bd} is the bandwidth-delay product of the end-to-end link. We investigated managing TCP connections using information on available buffer space at a base station to increase throughput, whilst maintaining end-to-end security.

6.4.5 Protocol

TCP allows each end of a connection to notify the other how much data it is currently willing to receive through a `window` header. Our TCP enhancement at the mobile host sets this value to the minimum of the TCP window and the buffer space available at the base station for the connection. This will cause the fixed host TCP sender to reduce the amount of data it is sending so that the buffer doesn't overflow.

While many base stations will allocate a fixed-size buffer to each client, the space available to each TCP connection will still vary according to the number of other client flows and local error conditions. Data can only leave a buffer when it has been successfully transmitted. High error rates will require data to be stored while it is repeatedly retransmitted. These error conditions can appear and disappear suddenly due to changes in the client's local environment – such as a passing truck. The number of other clients in each cell can also vary rapidly. For both these reasons, the variation in round-trip time of TCP connections is too high to simply rely on accurate TCP timers producing the correct data flow rate.

It is important that the mobile host has as fresh an estimate of the available buffer size as possible. Our modified base station stack therefore includes the available buffer size for a given client as an IP option header in a percentage of packets sent to that client. Using an IP rather than TCP option allows other transport protocols and encrypted IPSEC packets to be marked.

The percentage of packets marked with this information is determined by the amount and variability of buffer space available and the number of base station clients. Fewer clients and lower variability both reduce the rate at which the client-side information on buffer size becomes stale, and hence the need to use scarce bandwidth to send updates. Lower buffer space availability increases the urgency with which clients should be notified to reduce throughput to prevent packet loss.

The buffer size information will be slightly stale when it reaches the fixed host, as it will have travelled over the wireless hop to the mobile host and then right back over the connection before it takes effect. The mobile host therefore uses the rate of change of the buffer size to

estimate how much space will be free in one round trip time, the point at which the traffic from the fixed host will arrive at the base station. This is computed using:

$$estimate = current\_value + \frac{rtt(current\_value - last\_value)}{time\_difference}$$

(6.1)

where *current_value* and *last_value* are the current and most recent free buffer sizes from the base station, and *time_difference* the time elapsed between receiving those two values. An exponential weighted moving average can include more history information by including a part of the previous estimate, as used by TCP to estimate round-trip time:

$$estimate = \alpha(current\_estimate) + (1-\alpha)\, previous\_estimate, \text{ where } 0 < \alpha < 1$$

(6.2)

A higher α reduces the influence of previous estimates.

When the wireless link's throughput is reduced due to errors invoking link-layer retransmission, the data arriving at the base station will continue at the previous rate for at least one round-trip time between the fixed and mobile hosts. Therefore the buffer will overflow and packets will be lost if:

$$db_1 > db_2 + s$$

(6.3)

(where `d` is delay, b_1 is previous bandwidth, b_2 is current bandwidth and `s` is free buffer space.)

To avoid unnecessary complexity at the base station, the mobile host is responsible for allocating its available buffer space between different flows.

6.5 Reliable multicast enhancement

One of the most controversial issues in designing a reliable multicast protocol is ensuring its congestion-control behaviour is TCP-friendly. The utility of a flow to multiple recipients, amount of information required by the network, and efficient pricing models are all the subject of continued debate.

Crowcroft suggested that such protocols should have the following properties:

1. The rate of TCP unicast flows over a congested link should never be reduced to zero over a reasonable lifetime by a reliable multicast protocol.
2. The rate of a reliable multicast protocol over a congested link should never be reduced to zero over a comparable lifetime by TCP unicast flows.
3. Congestion control feedback traffic must be bounded to some small percentage of the data rate [Crowcroft01].

An example of a protocol meeting these requirements is `pgmcc` [Rizzo00]. It attempts to use feedback from the group member with worst connectivity to limit throughput to the rate at which a TCP stack would send data to that recipient.

With any such scheme that mimics TCP behaviour, wireless hosts can use the information they receive from our protocol on available base station cache to manage throughput from the sender. This has even greater benefit than with unicast, due to the overhead involved in retransmission of packets in reliable multicast protocols. Typically, retransmission requires work by either all other members of the group, or all of the aggregation nodes between the sender and recipient.

Mobile hosts can also use cache information to monitor how heavily loaded the local base station is. When reliable throughput is important, they may choose to switch to another base station in range that is less likely to lose packets.

6.6 Fast handoff

As mobile hosts move from one cell to another in a wireless network, they must transfer connections from their old to new base station. This handover must occur quickly if interactive applications are using the link.

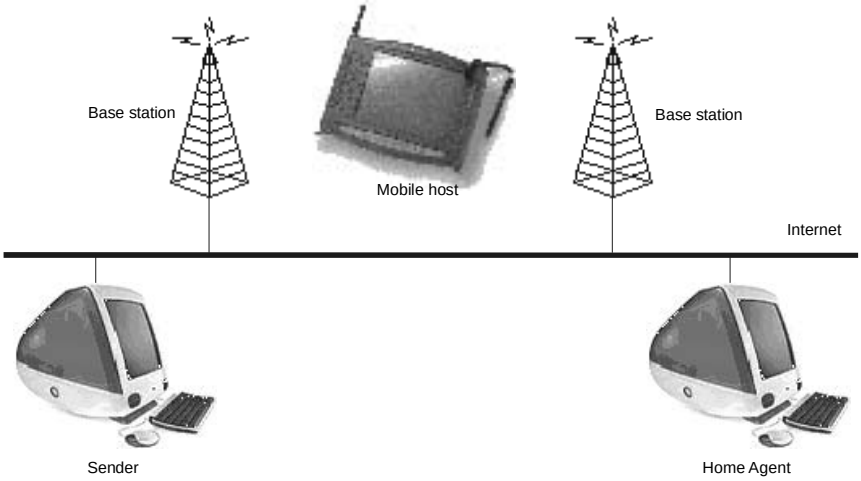


Figure 6.2: moving between base stations

A major problem with many schemes that provide services to mobile hosts at the wired/wireless base station boundary is the amount of state required to transfer connections using those services to a new base station. The Indirect-TCP scheme, for example, must move up to 32K of buffer for each TCP connection it is managing to the mobile host's new location. In one experiment, this increased handover time by up to 1430ms [Bakre94].

But even without these problems, handoff can cause packet loss and delay. Standard Mobile IP [Perkins96] routes IP packets to mobile hosts via a *home agent* on their home network that tracks their location and forwards packets appropriately. A *foreign agent* on the mobile host's local network receives these packets and sends them on locally. When an MH moves into a new cell, it must inform the foreign agent of its location. Until the foreign agent receives this information, it will continue to forward packets to the MH's old base station, which can no longer reach the MH. Either these packets will be lost, or the old base station will forward them to the new, causing delay and reordering.

Seshan et al. used multicast to ameliorate these difficulties. Their home agent forwarded packets to a multicast group that the MH asked each base station within range to join. The nearest base station forwarded packets from the group to the MH, while the others stored up to 12 packets in a circular buffer. When the MH moved to a new cell, it requested that cell's base station start forwarding packets, and the old base station stop. Because the new base station already had the most recent packets destined for the MH in a buffer, it could immediately forward them to the MH, preventing loss or delay [Seshan97].

6.6.1 Protocol

Our protocol is similarly designed to allow fast handover. But rather than multicast from the home agent, we use the foreign agent to multicast packets to the base stations, which either forward or buffer them as in Seshan's scheme. This is for three reasons:

- Multicast routing protocols are currently immature and work best either in dense mode where there are many members within a small section of the network, or sparse

mode where there are few members spread throughout the network. Seshan's system requires a mix of both: a sparse link between the home agent and base stations, but dense links between the base stations. Sparse mode protocols do not work efficiently with many group members joining and leaving frequently, as is the case where a mobile host is on the move and constantly changing the set of base stations within range. We use unicast to transport packets between the home and foreign agents, then a dense mode multicast protocol that can better support a changeable group membership.

- A mobile IP extension [Perkins99] allows hosts to send packets directly to a MH's foreign agent rather than via its home agent. This reduces latency and traffic if the home agent is outside the most direct link between the host and MH. This is not possible if a single-source multicast protocol like PIM Express [Holbrook99] is being used to transport packets from the home agent to the MH's base stations.
- This better fits our security model. It forces all packets destined for mobile hosts on a given network to be routed through the network's foreign agent(s), which provides a point of control for the network operator. It also allows location privacy, hiding the location of the mobile host from even its home agent, if an anonymous routing protocol such as Onion Routing [Reed98b] is used to link the home and foreign agents. Multicast groups cannot have anonymous members, because multicast routing protocols rely on accurate location information to prevent loops.

We aim to slightly update the mobile IP standard to provide this forwarding functionality, by allowing a multicast address to be specified for the MH. The MH itself does not join this group, but instructs all base stations in range to do so. Base stations decapsulate packets as they arrive from the foreign agent then forward or buffer them appropriately.

Rather than impose onerous synchronisation requirements on base stations, we allow a packet to be lost in the unlikely event it is removed from the buffer of the mobile host's new base station before handover is complete. In this case we rely on TCP's retransmission algorithms to repair the loss. The small resulting reduction in throughput is a reasonable trade-off against the extra complexity, communications bandwidth and buffer space required to produce a fully reliable rather than *resilient* protocol [Balakrishnan95].

6.7 Conclusion

We began this chapter by describing the mechanisms used to provide reliable data transport over the Internet. TCP's retransmission and congestion control mechanisms have proved to be a successful way to build reliability and link-sharing onto an unreliable network in an end-to-end way.

We then described the problems caused for TCP by error-prone wireless links, and previous suggestions for reducing the sometimes disastrous effects these can have on throughput. *Indirect-TCP* terminates a TCP connection at the base station and uses a wireless-optimised protocol to transport data over the last link. But it removes the essential end-to-end reliability expected by applications, as well as any transport or network-layer security protection. *TCP snoop* maintains end-to-end reliability and produces impressive increases in throughput, but is incompatible with the network-layer security that will be ubiquitous once IPv6 becomes widespread. Compatibility would require *snoop* access to the cryptographic keys being used to authenticate and encrypt traffic, removing end-to-end security.

We therefore developed a protocol that uses link-layer reliability functions whilst preventing damaging interactions with TCP's retransmission features. Base stations notify mobile hosts of available layer two cache size, which use this information to reduce TCP throughput when required to prevent segment loss. We also described how mobile hosts participating in a reliable multicast protocol could use cache information in controlling throughput in a group to reduce the incidence of packet loss and choosing between available base stations.

Finally, we showed how fast TCP handover could be provided by modifying an existing protocol to increase security and make better use of available multicast routing protocols.

7 Distributed fingerprinting

#define m(i)(x[i]^s[i+84])<<

unsigned char x[5],y,s[2048];main(n){for(read(0,x,5);read(0,s,n=2048);write(1,s ,n))
if(s[y=s[13]%8 + 20]/16%4==1){int i=m(1)17^256+m(0)8,k=m(2)0,j=m(4)17^m(3)9^k *2-
k%8^8,a=0,c=26;for(s[y]-=16;--c;j*=2)a=a*2^i&1,i=i/2^j&1<<24;for(j=127;++j<n ;
c=c>y)c+=y=i^i/8^i>>4^i>>12,i=i>>8^y<<17,a^=a>>14,y=a^a*8^a<<6,a=a>>8^y<< , k=s
[j], k="7Wo~'G_\216" [k&7]+2^"cr3sfw6v;*k+>/n." [k>>4] * 2^k * 257/8, s[j]= k^(k&k *
2&34) * 6^c+~y;}}

–Charles M. Hannum

7.1 Introduction

The communications security schemes described in chapter two all protect information as it is transmitted between sender and receiver. But sometimes a sender wishes to keep track of data beyond that point. The Internet allows content to be retransmitted discretely on a large scale. Having sold valuable content to a client, sources do not want clients to be able to re-sell it on to others.

Watermarking allows content providers to embed virtually undetectable information into data such as audio and video streams. If these marks are unique to each recipient, this process is called fingerprinting. If a subscriber illegally redistributes such data, it allows the provider to trace any pirated materials they discover back to the originator, and take legal action to recover any losses they have suffered through those materials. But the problem with using fingerprinting with broadcast or multicast media is that the addition of distinguishing marks to the data stream is contrary to the bandwidth saving of being able to distribute the same data efficiently to multiple recipients.

This chapter describes a system we have developed that allows plaintext *or* ciphertext media to be efficiently fingerprinted by a source and active network components in a multicast tree. Our method builds upon some of the mechanisms proposed to provide support for reliable multicast data delivery and for delivery of scalable, layered, streamed multicast data (such as video and audio), as described in chapter three.

In the next section we describe related work on content protection. We then outline the salient points of our approach and analyse its efficiency and possible threats to its effectiveness.

7.2 Content protection using trusted hardware

The traditional approach to redistribution control relies on prevention by trusted client hardware. The most recent example of such schemes, and their problems, is the Digital Versatile Disk system. DVD-Video player manufacturers must accept a license agreement that imposes several requirements on their hardware to restrict media redistribution [Bell99]:

- Digital video outputs are only allowed over an encrypted channel to another compliant device. Analogue outputs must include an approved analogue protection scheme like Macrovision that prevents analogue VCR copying.
- Watermarks in the data specify whether copies may be made of that data. Compliant display devices must not play unauthorised copies.
- A Content Scrambling System (CSS) cipher is used to encrypt data on disks. Only licensed manufacturers are given one of the 408 master keys needed to decrypt the session key on each disk, which enables decryption of the disk content.

Unfortunately for the DVD Forum, this model has proven to be rather flawed.

CSS is a poorly designed cipher. It uses 40-bit keys, which can be trivially discovered through brute force search. This was a decision made to allow global export of players under the

contemporary US export controls – even though the Bureau of Export Administration would likely have allowed stronger ciphers given that the DVD Forum could have provided them with the 408 master secrets allowing decryption of any data. But an attack has anyway already been discovered that allows ciphertext to be decrypted with a workload of only $O(2^{16})$ [Stevenson99].

The master secrets in players are starting to be reverse engineered and put into software DVD players. One such program, DeCSS, has become the target of ferocious legal activity by the Motion Picture Association of America. They have obtained an injunction in the New York courts against websites hosting the program, and encouraged the Norwegian police to arrest and confiscate the computing equipment of a 16-year old Norwegian alleged to have authored the program [Burke00]. The DVD Copyright Control Association obtained an injunction in California against sites publishing the program, master keys or CSS algorithms [Elfving00]. Unfortunately they introduced the source code for CSS, the trade secret at issue in their case, as a public court document. By the time they realised their mistake and had the document sealed, just one site hosting a copy had already reported 70,000 downloads [Young00].

But most simply, the entire security apparatus can be circumvented through basic bit-wise copies of DVD disks. While blank disks are currently sold with a crucial header area pre-embossed, preventing pre-recorded disks being copied onto them, it will take very little effort for pirates to obtain truly blank disks. Thus large-scale piracy will flourish, while consumers are denied their fair rights with regard to making personal copies [Samuelson98].

Such vulnerabilities will be present to a greater or lesser extent in any scheme that relies on a client acting against the interests of its owner (the consumer). The amount of effort an attacker will expend on defeating such measures is a function of the value of the protected material and how easily it could be re-sold. Because Internet transmissions have virtually zero marginal cost, re-sale is very easy and hence an attacker has a great incentive to defeat even complex schemes protecting any kind of valuable content

Smartcards are often mistakenly considered to allow protocols to act against the interests of their owner, due to their “tamper-proof” protection of stored keys. Unfortunately this property is better described as “slightly tamper-resistant.” A wide spectrum of attacks, both physically on cards and using analysis of card power consumption from an external line, make it simple for a determined adversary to compromise current cards [Kömmerling99].

Cryptoprocessors, microprocessors that can decrypt and execute code on-chip [Best80], have been touted as a means of preventing reverse engineering and secret recovery from software. They provide an extremely high level of assurance against software tampering. But again, they would rely on a small number of master secrets held by chip manufacturers used to sign chip certificates. If compromise of one of those secrets allowed pirate microchips to be produced, there would be an enormous incentive for attackers to do so.

There are also grave civil liberties issues with allowing a chip manufacturer rather than the owner of a machine to control which software is allowed to run. The behaviour of the US government described in chapter four provides a template for the impositions it would be likely to force upon chip manufacturers in any way vulnerable to the US legal system. Preventing the execution of strong encryption software could be only the beginning of restrictions [Gladman99].

The few large-scale systems that have used this type of technology have also proven unpopular with consumers. DivX (Digital video express) was a video format that allowed users to watch a newly purchased title for two days before needing to pay for further screenings. Encryption was used to protect content, and players would only display content that had been paid for. Consumer dislike of these features led to the death of the format [Kane99].

7.3 Broadcast protection

Typically, in broadcast networks, cryptographic techniques are sufficient to give an acceptable level of protection against dishonest reception. Pay-TV systems use Entitlement Control Messages (ECMs) embedded in transmissions to control access by subscribers to video they are entitled to view. The ECMs are interpreted by a smartcard in the subscriber's set-top box, and result in a stream of keys from the card that allow the box to unscramble the audio and video streams [Anderson01].

The major problem is legitimate users illegally selling their keys to others. Second-generation pay-TV protection schemes used the same keys for every subscriber authorised to view a given programme. So anyone who could insert a PC between a smartcard and set-top box could intercept and then re-sell that stream of keys, allowing encrypted content to be recorded and later decrypted by non-subscribers [Anderson01].

Schemes have been designed to make keys effectively as large as the content they protect, and hence uneconomic to sell [Dwork96], or to identify the source of stolen keys [Naor88] even when legitimate users collude to try and cover their trail. These schemes are no protection against retransmission of content rather than keys, but are targetted at an environment where retransmission of content is difficult and expensive.

In the Internet environment, retransmission of content is simple and cheap so further protection is essential.

7.4 Efficient fingerprinting

Fingerprinting allows content providers to trace any illegally redistributed data they discover back to a subscriber by subtly marking the data sent to each user in a way that is difficult to reverse.

The simplest fingerprinting schemes use the low bits in an audio or image file to embed information such as subscriber ID. These are easily defeated by altering these bits. More complex schemes select a subset of bits to alter using a secret key, or use spread spectrum techniques or complex transformations of the data to make removal of the fingerprint more difficult.

Anderson and Manifavas describe an ingenious scheme that allows a single broadcast ciphertext to be decrypted to slightly different plaintexts by users with slightly different keys. Unfortunately the scheme is extremely vulnerable to collusion between users. Five or more users can together produce plaintext (or keys for installation in pirate decoders) that cannot be traced. Shamir has pointed out that increasing collusion resistance in all of these schemes requires exponential work from the defender to cost the attacker linearly more effort [Anderson97].

Regrettably this is the only scheme proposed so far that allows efficient marking of data being supplied to large numbers of users. While others are not hugely computationally intensive, they would not scale well. A content provider would need enormous computing resources to be able to fingerprint data being sent to typical live event audiences. Microsoft estimated that 10.8 million viewers watched its webcast of Madonna's London concert on 28 November 2000 [MSN00]. An enormous amount of bandwidth would also be used up sending a different version of the data to each viewer. The Madonna webcast was streamed at a minimum of 56kb/s. If unicast, this would have required processing and transmission capacity of 560Gb/s. This motivates our approach, which distributes the processing needed throughout the multicast tree used to efficiently deliver data.

7.5 Multicast Security

The IP multicast model allows for *any* receiver to become a sender, subject to their ISP's policy and pricing mechanisms.

Unlike traditional broadcast networks, the effort needed to re-multicast data is small. Further, any host can receive multicast traffic, with the decision to route data to that host being made close to that host with no reference to the original source of the data. It becomes clear that, in addition to marking data to trace those who illegally redistribute it, we also need to protect data in transit to prevent unauthorised access by eavesdroppers.

By encrypting packets, we ensure only those possessing the necessary key can access content. As described in chapter two, this can most usefully be performed at the application and network layers. Whilst the problem of key management for multicast data is not yet completely solved, proposals to provide this functionality [Hardjano98a] are moving forward and could build on content providers' systems for authenticating users. With rapid re-keying, content providers could remove pirates from groups even quicker than current pay-per-view systems.

It is possible to limit multicast traffic to a specific region of the network, using administratively scoped addressing [Meyer98]. This relies on border routers of the administrative region being correctly configured to prevent traffic sent to certain address ranges leaking out of the region. It provides an effective means of limiting the flow of traffic if correctly configured, but does not prevent unauthorized reception of data by hosts within the region. It is also difficult to configure and use, although future protocol developments may ease these problems [Handley98b].

It is almost impossible to limit access to multicast data purely to authorised receivers. We must rely on encryption and good key management to prevent intercepted traffic being decoded, and fingerprinting to trace authorised users who illegally redistribute content.

7.6 Router Support

Reliably multicasting a packet to a large group of receivers becomes more efficient if the network acts to ensure reliability. A number of proposals have been made to add such reliability into the network. One of these, PGM [Speakman99], uses Negative AcKnowledgements from packet recipients, summarised by routers as they travel back up the distribution tree, to trigger retransmissions from senders. Routers forward the retransmitted packets to those recipients who reported them lost. Thus a separate multicast tree is built for the retransmission of each packet. Routers also multicast a NAK ConFirmation on the local subnet that originated the NAK. Any host that possesses the data referred to by a NCF may retransmit the packet locally.

PGM provides a close fit for our requirements for a fingerprinting scheme. It offers a number of end-to-end options to support fragmentation, sequence number ranges, late joins, time-stamps, reception quality reports, sequence number dropout and redirection. Of interest to us is the sequence number dropout option. This allows placement of "intermediate application-layer filters" in routers. Such filters allow the routers to selectively discard data packets and convey the resulting sequence number discontinuity to receivers such that sequencing can be preserved across the dropout, and to suppress NAKs for those packets intentionally discarded. They act as lightweight active network elements, modifying data streams passing through them. The operation of these filters is not defined by PGM. In later sections of this chapter, we describe semantics for these filters suitable for fingerprinting multicast streams.

7.7 Layering and FEC

The use of packet-level forward error correction data to recover from loss is well-known. For every k data packets, $n-k$ FEC packets are generated, for the transmission of n packets over the network. For every transmission group of n packets it is necessary to receive only a subset to reconstruct the original data.

There are a number of means by which these FEC packets may be transmitted. The three primary means are by piggy-backing them onto previous packets, sending them as part of the same stream but with a different payload type indicator or sending them as a separate stream.

Sending FEC packets within the same stream as the original data has the advantage of reducing overheads (routers only need keep state for a single stream), but forces all receivers to receive the FEC data in addition to the original data. If a receiver is not experiencing loss, this is clearly wasteful.

Sending the FEC data on a different stream has greater overhead (because routers need keep state for multiple flows), but allows for greater flexibility. Those receivers which are not experiencing loss do not join the multicast group transporting the FEC stream, and hence do not receive the FEC data; varying amounts of FEC can be supplied, layered over a range of groups, giving different levels of protection [Perkins98].

7.8 Protocol Overview

The *loss signature* of a receiver is used by a number of multicast protocols to identify subsets of receivers that belong, at least symptomatically, to shared subtrees. Studies have shown a large amount of heterogeneity in the reception quality for multiple receivers in a single session. Many receivers see the same patterns of loss due to lossy links in the distribution tree that cause loss for all child nodes beneath them [Yajnik96].

Given that the loss signature of a receiver corresponds to its position in the network, it should be possible to use this as a simple form of digital fingerprint. The pattern of degradation in a stream will likely be different for each receiver provided there is a non-zero packet loss rate in the network. There are four problems with this:

- A receiver may neglect to send a loss signature back to the sender, escaping notice by the fingerprinting scheme.
- Lost packets cause degradation of the delivered stream. A network that drops enough packets to make this fingerprinting technique successful will likely provide insufficient quality for most uses.
- A receiver may collude with another receiver to repair the loss, hence defeating the fingerprinting scheme.
- A receiver may easily defeat the fingerprinting scheme by dropping additional packets (possibly transforming the stream to match that of another receiver).

Ensuring that a receiver returns its loss signature to the sender is clearly an impossible task in the traditional Internet environment with smart end-points and dumb routers. However, if an active network is assumed it becomes possible for the last-hop router to return a loss signature to the sender. If the installation of this active element forms part of the multicast tree setup procedure, we may ensure that the loss signature of each receiver is returned to the source.

The active network elements can also conspire to ensure that all receivers see unique loss patterns, rather than leaving this to chance. Instead of relying on the loss signature of a particular branch in the multicast forwarding tree being unique, the position of a node in the tree can be used to determine which packets to drop in order to ensure a unique loss pattern for each node.

The proposed use of active network components is not unique to our scheme; a number of reliable multicast protocols have been developed which would benefit from support within the network. This support typically takes the form of filtering, summarisation and subcasting abilities: exactly the requirements for our scheme.

The assumption of an active network leaves three barriers to the development of an effective fingerprinting solution: degradation of the stream by packet loss, collusion attacks by multiple receivers to repair the stream, and the ease of breaking the protection by dropping additional packets. These three problems are related, and have a common solution: the source produces multiple versions of each packet, with differently watermarked contents.

This altering of the media stream is typically straightforward, although content specific. It is vital that a set of transformed packets resulting from one packet cannot be used to recreate the original, otherwise a collusion attack could produce a non-fingerprinted version of the data. Likewise, the fingerprint must be resistant to a wide range of transforms, such as the introduction of jitter or re-sampling [Petitcolas98].

The active network elements then *subtract* packets. Rather than ensuring a unique loss pattern at each receiver, they ensure a unique pattern of packets is received. This may be implemented using the PGM sequence number dropout option and application layer filters as previously noted.

This solves the quality degradation problem, since each participant receives one version of each packet. The reception quality is no worse than that provided by the underlying network, although each receiver sees a slightly different stream. Receivers can no longer collude to repair a stream (the result will simply be a combination of their fingerprints, enabling identification of the conspirators). Finally, discarding additional packets simply results in a degraded stream with the fingerprint still present.

The result is a relatively simple means of fingerprinting multicast data: the source sends multiple subtly different copies of each packet. Routers at the branch points in the network discard packets, such that the stream delivered to each receiver is unique. We call this combination of watermarking and multicasting *watercasting*.

7.9 Protocol

We have designed an initial protocol to implement watercasting using PGM and slight modifications to multicast tree setup. These could be made using active network code in routers. We hope to refine this protocol after gaining implementation experience.

A client wishing to receive content from a server first performs a unicast authentication with that server. After convincing the server it is a valid subscriber, the client is given a receiver identification key.

This key is supplied to the last hop router when the receiver joins the session. The last hop router passes this key, its address and the time the receiver joined back up the multicast tree to the source. Each router in the tree adds its address, encrypted with the public key of the server to keep the topology secret. This is a slight variation on the current Internet Group Management Protocol MTRACE packet. When the source receives a valid RIK and topology report, it unicasts the current session key(s) for the requested media to the client.

This server is therefore able to validate and store the entire tree topology. This information is necessary to allow it to later determine the correct fingerprint for each receiver, in the event that it becomes necessary to trace an illegal copy of the content.

For a multicast distribution tree with maximum depth d, the source generates a total of n differently watermarked copies of each packet such that $n \geq d$. Each group of n alternate packets is termed a transmission group.

On receiving the packets which form a transmission group, a router forwards all but one of those packets out of each downstream interface on which there are receivers. The choice of which packet to discard is made at random on a per-interface basis, with the pseudo-random sequence keyed by the position of the router in the distribution tree and the interface address.

Each last hop router in the distribution tree will receive $n - d_r$ packets from each transmission group, where d_r is the depth of the route through the distribution tree to this router. Exactly one of these packets will be forwarded onto the subnet with the receiver(s). The choice of which packet is forwarded is determined pseudo-randomly, keyed with the position of the router in the tree, the interface address and the receiver identification key.

The filtering process is illustrated in figure 7.1. In this example, the receiver furthest from the source is R1 and the maximum depth of the distribution tree is $d=5$. The source will generate $n \geq 5$ distinct versions of each packet to form a transmission group; label these ABCDE. At router 0 these are filtered, passing ABDE to router 00 and ACDE to router 01. At router 01 the packets are filtered again, with ACE being passed to router 010. Since this is the last hop router before receiver R2, router 010 does not just filter out a single packet, it pseudo-randomly selects one packet, E, to pass to the receiver. A similar process occurs to filter the packets destined for the other receivers.

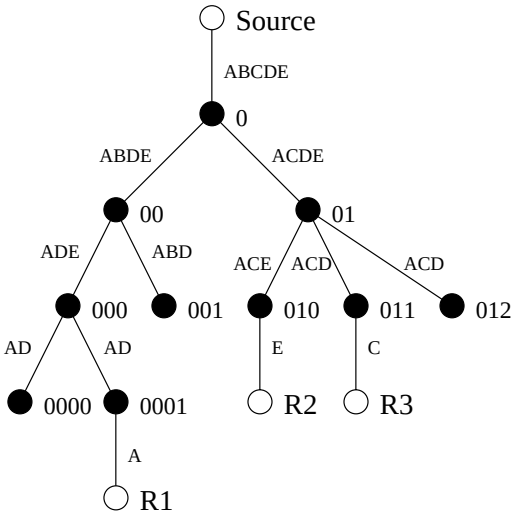


Figure 7.1: Filtering transmission groups to obtain a unique fingerprint

The media payload is encrypted. The receiver also receives a decryption key from the source by the same means that it receives the receiver identification key. Media packet headers are not encrypted, since routers need to use the sequence numbers in the packets to determine which packets to discard.

The encryption of the media payload prevents unauthorised receivers from snooping on the packets. The fingerprint may be used to detect illegal redistribution of the decrypted payload by legitimate receivers.

In effect, the combination of tree topology and receiver identification key is the secret used by the source to do the fingerprinting. Participating routers should therefore refuse requests to reveal any part of that topology. But even if some routers and clients collude, they would need a conspiracy from a client right up to the source to discover anything useful.

The selective discard function aims to provide the multicast routers and their clients the minimum degree of freedom possible in order to facilitate the later tracing of cheating routers or users. Every router in the tree indelibly affects the stream by dropping certain packets. A cheating downstream router therefore cannot produce a stream seemingly originating from an upstream router, as it does not have access to all of the packets passing through that router. It would need to collude with all other branches of that router to get copies of all data passing through it. The higher up the tree, the less likely this is, the lower down the tree, the easier to eliminate targets from an investigation. An upstream router could attempt to impersonate a downstream router or client, but would need to know the topology of the tree to that point to do so effectively. This becomes increasingly difficult as the distance of the point from the cheating router increases, which may be sufficient protection in that points upstream of a suspected router could be included in any investigation of that router. Alternatively, each router could be given a shared secret by the source at the time it joins the multicast tree, and include that secret in the initialisation of its pseudorandom number generator.

We keep as much of the processing as possible at the source to simplify the router protocol. For each watercast stream a router is processing, it only needs to store the sequence state to allow it to drop the appropriate packet in each transmission group. We place a heavier burden

on the source: it needs to appropriately modify the outgoing stream and inject sufficient redundant packets to maintain quality for all clients whilst allowing enough packets to be dropped to fingerprint each client's stream uniquely.

The source also needs to store enough information to enable it to later reconstruct the path to a fingerprint found in a recovered media clip. Because the fingerprinting algorithms, and the pseudo-random sequence generator, operate in a deterministic manner, this comprises the original data, the topology of the distribution tree, and details of when receivers join and leave.

Given a recovered clip, the server can determine how many redundant packets it was sending out at that time and hence which transformations were being applied. It can then reduce the clip to a series of packet labels (such as AECDBBABC). By simulating the operation of various network components from the start of the transmission of the original broadcast through to the end of the clip, it can aggressively rule out nodes that could not have produced the clip because they did not have access to any of the packets present in it.

The completion of this process will result in a set of nodes that could have produced the given clip. The length of clip required for this result depends on the multicast tree topology and pseudo-random sequence generator used. This can vary continuously according to the topology of the multicast tree at the time of transmission of the marked data, something known only to the source. This makes it difficult even for conspiracies of users and routers to remove the fingerprint or alter it to implicate someone else. The probability that the media clip must have originated with a particular receiver increases with the length of the clip.

7.10 Analysis

The two important effects of watercasting are reducing the number of unique watermarked copies of data required, and distributing the task of selecting a different sequence of watermarked packets to each recipient. The results of these gains are considered below.

The probability that a particular version, v, of a packet is received, given that the transmission group size is n and the receiver is d hops from the source, p, is simply

$$p_v = \frac{1}{n-d+2}\prod_{i=1}^{d-2}\frac{n-i}{n-i+1} \equiv \frac{1}{n} \qquad (7.1)$$

The probability that multiple receivers receive the same version of a packet depends on the position of those receivers in the distribution tree. The closer those receivers are, that is the longer the shared path from the source, the more likely they are to receive the same version. The probability that two clients A and B will receive the same version of a packet is obtained by replacing n in equation 7.1 with $n-(h-1)$, where h is the number of hops from the source to the last router on the shared path to A and B:

$$p(A_v = B_v) = \frac{1}{n-h+1} \qquad (7.2)$$

If cooperating attackers are topographically close, they will find it more difficult to obtain the different versions of each packet that are necessary to try and disguise the recipients of a given stream. Even if a number of conspiring attackers manage to successfully produce a disguised stream, it will be much harder for them to hide their general location in the network. They will almost certainly be traceable to a given organisational or regional sub-net.

But ideally, any two given clients' streams will vary enough to enable their differentiation, but by too small an amount to make any useful disguise by combining their streams. If a is the number of attackers in a collaborative conspiracy, and l is the length in packets of the

stream they want to disguise, the probability they can obtain every version of each packet is a simple extension of the coupon collector's test [Knuth69 §3.3.2]:

$$p = \left(1 - \frac{n!}{n^{a-1}} \left\{ \frac{a-1}{n} \right\} \right)^l, a \geq n$$

(7.3)

where $\left\{ \frac{a-1}{n} \right\}$ are Stirling numbers [Knuth69 §1.2.6].

The $\frac{n!}{n^{a-1}}$ term shrinks rapidly as $a >> n$, increasing *p*. Increasing the value of *n* therefore provides an effective defence against collaborative conspiracies. And equation 7.3 is a best-case scenario for attackers, where each conspirator has an equal chance of receiving a given version of a packet e.g. there are no shared routes between the attackers. Shared routes only increase the probability than attackers will receive the same version of a packet, making conspiracies more difficult.

The tradeoff of increasing the size of a transmission group is the extra bandwidth used. Introducing redundant data into the multicast stream increases the size of the stream by the number of packets per group at the first hop, then one less at the second hop, and so on until the last hop where the traffic size is the same as for a non-fingerprinted stream. If *n* is the group size and *d* the maximum depth of the tree, this increases the amount of traffic by a factor *f* of

$$f = \frac{n}{d} + \frac{1}{d} \sum_{i=1}^{d-2} (n-i) + \frac{1}{d} \equiv \frac{2n - d + 3}{2} - \frac{n}{d}$$

(7.4)

This is still far less than the traffic that would be generated by unicasting unique versions of each stream to every receiver. If we set $n=d$, which creates the minimum extra traffic but makes the fingerprint sequence longer, this factor is

$$\frac{d+1}{2}$$

(7.5)

At the cost of greater complexity at routers, this figure could be decreased still further. If each router knows the maximum depth of each of its interface's subtrees, it need only send the minimum number of redundant packets necessary to each child node. Rather than choosing one packet from each transmission group to drop at random, it selects n-d_i packets to drop, where d_i is the depth of the subtree on that interface. This reduces the extra bandwidth consumed on all subtrees that are shallower than the maximum depth of the tree. It also reduces the scope of attacks by cheating downstream routers, which have fewer versions of each packet to use.

Using a value of $n > d$ allows a tree to grow more easily. At the initial setup of a tree, this is particularly important: as many new members join, continually altering the depth of the tree, the source would otherwise need to constantly increase n. By setting n to the likely depth of the tree after this phase, the source reduces this update complexity, at the cost of greater bandwidth requirements over the (initially small) tree as it is set up. The source may use knowledge of other likely tree changes, or react to a rapidly-changing depth, by discontinuously varying *n* in the same manner.

There is therefore a tradeoff: larger values of *n* allow greater flexibility in tree setup and reduce the length of the fingerprint sequence required to trace the originator of data, but require greater processing and bandwidth to create and distribute.

An obvious optimisation would seem to be only fingerprinting 1 in every *x* packets, reducing bandwidth and computation requirements by a factor close to *x*. But because every router in the multicast tree would need to know the location of the fingerprinted packets to run the watercasting algorithm, it would be impossible to keep this information secret.

Unfortunately, if an attacker knew the position of the fingerprinted packets, she could simply remove them and redistribute the resulting degraded data. While this would be fatal to data such as executable code, it may be acceptable for lossy information such as an audiovisual signal. An information provider must consider the quality of data they are effectively prepared to give away before using this technique.

When the last-hop network is a multi-access subnet, such as an Ethernet, any host on that network can receive the same packet with no extra effort. Non-subscribers on a network can intercept such packets, but do not have the decryption key needed to read them. But two legitimate subscribers on the same sub-network will receive the same fingerprinted data. We contend that this is a small problem: multi-access sub-networks are typically under the administrative control of a single agency. If one of the users of such a network illegally resold data, it would be traceable to the agency controlling that network, which is sufficient in most cases.

Collaborative conspiracies are always a difficult problem for fingerprinting schemes. Groups of users can attempt to combine their different fingerprinted versions of the same piece of data in a way that removes or at least damages the fingerprint. The simplest way to do this is perform 'bit voting': set each bit in the reconstructed piece of data to be that which is most prevelant in the same bit in the set of fingerprinted files. This is usually fatal to simple schemes, and can damage more sophisticated fingerprints.

The fingerprinting techniques we use must therefore be able to defeat collaborative and individual attacks such as introducing jitter and re-sampling [Petitcolas98]. But the main contribution of this chapter is that an active network can perform part of the fingerprinting function. Even if the specific transforms we use can be defeated, we hope that it should be possible to simply plug in others more resistant to attacks, preserving the validity of our approach.

7.11 Conclusion

We have outlined a general idea for fingerprinting media data using active network components compatible with end-to-end security, and a specific method to perform that task. Our method leverages schemes such as PGM that are being developed to provide other services such as reliable multicast and filtering in active networks.

Traditional communications security systems protect data *en route* to recipients by encrypting it with a key known only to authorised users. This is effective at preventing eavesdropping of the data in transit, but can do nothing to stop authorised recipients redistributing the data. This is a major problem for 'secrets' such as live sporting events shared between millions of paying customers. As the cost of redistributing data continues to plummet, fingerprinting is likely to become as essential to information security as cryptography is today.

In conventional fingerprinting schemes, each recipient gets a different version of the fingerprinted data to allow any cheating recipients who illegally redistribute the data to be traced. This is computationally expensive for the source, which needs to calculate a large number of unique versions of the data, and requires unicast transmission of the resulting data to clients. Our scheme reduces both loads. The source needs to calculate a far smaller number of versions of each packet of the data, and is relieved of the task of deciding which packet goes to which user by routers. This load is spread thinly throughout the distribution tree. The

resulting data can be transmitted efficiently through the network at a cost in bandwidth over pure multicast related to the depth of the multicast tree rather than the number of recipients. This is particularly important for very large multimedia streams.

Watercasting requires a considerable amount of network complexity compared to content control schemes such as Nark [Briscoe99]. That uses a trusted smartcard to provide the keys needed to decrypt data according to an access policy, thus scaling well even with a constantly changing membership. Watercasting is more appropriate for protecting high value content where sufficient incentive exists for an attacker to compromise a smartcard.

Watermarking technology is still in its infancy. Petitcolas *et al.* [Petitcolas98] hoped that their attacks on first-generation algorithms would lead to an improved second generation, and so on. We hope our system is reasonably resistant to the attacks they designed, but we will no doubt see further ones developed. It is therefore important that any deployed system is easily upgradeable.

Watercasting has wide applicability to the protection of any data that is distributed to a large number of people via a network. While we have focussed on audiovisual data, other information such as software could be equally covered with the development of appropriate transforms. Indeed, an appropriate software architecture would allow small pieces of transform code to be plugged-in to our system to extend it with minimum effort.

The authenticity of such data may be more important to clients than that of a video broadcast. It would be trivial to use public-key authentication and so assure clients of the information's integrity and origin. Servers authenticating large amounts of dynamically generated data could use a hybrid scheme combining public-key certificates and *k*-time signature schemes that allows offline pre-computation of expensive parameters so that even bursty, lossy and low-latency streams can be authenticated [Rohatgi99].

We believe that the best use for our system is in the transmission of 'live' data. As Barlow observed [Barlow94], the value of such transmissions drop rapidly as they age. The live TV rights to a popular sporting event are worth a considerable amount, but this drops dramatically once the game is finished and the result is known.

Therefore, even if our fingerprinting scheme can be defeated, as long as it takes a reasonable amount of time to do so it will have achieved its main objective – to prevent large profits being available from the illegal re-distribution of content.

8 Conclusions and further work

"Law enforcers... are used to robbers and guns. There are now new criminals out there that don't have guns. They have computers and many have other weapons of mass destruction."

–Attorney-General Janet Reno

"If business wants to jump onto the surveillance and police state bandwagon, they should have to do it with their own resources, including their own standards planning, their own insightful and clear-thinking designers, and their own money. With luck, they will come up with something as clear and easy to implement as the SET standard."

–Thomas Junker

"We should not be building surveillance technology into standards. Law enforcement was not supposed to be easy. Where it is easy, it's called a police state."

–Jeff Schiller

"I have not given up on encryption."

–FBI director Louis Freeh

We began this thesis by examining the state of the art in secure Internet communication. There are a multitude of security protocols available at various layers of the protocol stack, many with overlapping functionality. We compared and contrasted the most important protocols, and suggested that a combination of network-layer IPSEC and application-layer assistance provides a simple yet flexible base upon which to build secure active service systems. But we cautioned that system designers must consider the entire path of data, including the first and last hop to users where necessary.

In chapter three we showed how this combination could be used to secure the various multimedia conferencing protocols. IPSEC with lightweight application-layer key management provides a secure group communication environment that is usable today. The IETF is beginning to develop multicast extensions to the Internet Key Exchange protocols that will provide an alternative to these application-layer systems, but it is likely to take several years before they are finalised and then deployed on a wide scale. We also described the multimedia coding and transport algorithms that are used in these systems as background to later chapters.

Chapter four examines the trust model implicit in almost all previous active network research: that intermediate points in the network should be allowed access to the plaintext of encrypted data flowing through them. We show that this poses unacceptable risks to the confidentiality and integrity of such data. Traditional threats such as hacking are being joined by an increasing number of unconventional attacks – from legal attacks on the authenticity and evidential value of data, to economic espionage using the "law-enforcement interfaces" being designed into the core of new networks such as 3GPP.

We then proceeded to see if it was possible to design systems that have the advantages of in-network processing whilst retaining the benefits of end-to-end security. Chapter five described the most commonly used active services today: proxies that reduce the bandwidth required by multimedia conferencing data. We showed that distributed packet filters allowed appropriately-coded video to be adapted to available bandwidth more efficiently and securely than current solutions, without reducing users' perception of the quality of the resulting video. We also demonstrated that these filters can be used to defend against distributed denial of service attacks with only minor modifications to critical points within the Internet infrastructure.

Next, we described another set of popular active services that improve the performance of TCP over lossy wireless links. These cannot work in the presence of implementations of the current IPSEC standard. Rather than require IPSEC modifications and the insecurity of

providing cryptographic keys or packet plaintext to the services, we described the design of an alternative protocol that avoids both, provides fast handover performance and assists reliable multicast protocols.

Finally, chapter seven described a new service that is equally functional operating on plaintext or ciphertext: distributed fingerprinting of multicast media. We showed that it is possible to retain the bandwidth efficiencies of multicast whilst uniquely marking each copy of data delivered to recipients. We believe that providing such audit trails will ultimately prove more successful in protecting copyrighted materials than the "trusted hardware" approach being pursued by most digital rights management companies at this time.

It is gratifying to note that system designers have now started taking note of these considerations. The Wireless Application Protocol originally ran a WapTLS-TLS converter at the gateway from WAP to IP networks [WAP98]. The security deficiencies of this approach, as outlined in this thesis, were caricatured as the "WAP gap". This has led banks to demand the replacement of this system with a tunnelling gateway that carries the original WTLS traffic over IP to its destination where it is decrypted [WAP01]. This is a simple yet effective outside confirmation of our central hypothesis: that active services can usefully operate on ciphertext flows *and* maintain end-to-end security. In this case, the only extra overhead is the WAP headers tunnelled across the wired network. WAP server operators have been happy to incur this small increase in data across well-provisioned networks in return for true end-to-end security.

8.1 End-to-end considerations

The "end-to-end" philosophy has been absolutely central to the design of Internet. From its core TCP and IP protocols through to its most popular application-layer protocols such as HTTP, no reliance is placed on functionality "within" the network. Protocol designers push as much required functionality as possible into smart end-points, leaving a dumb network in-between [Saltzer84]. The resulting scalability and robustness have been vital in maintaining the performance of the Internet during an exponential increase in its number of users. Even scholars far outside the computer science field now recognise that the flexibility resulting from the end-to-end philosophy has been vital in the rapid evolution and growth of the network [Lessig99].

Active services are contrary to this philosophy. There have been continued spirited attacks by Internet pioneers on the very idea of moving any functionality into the network [e.g. Reed98, Deering99]. Deering complains that routers should forward an incoming packet to one or more outgoing interfaces, and nothing more. But ironically, this is an extended definition he created to accommodate his pioneering work on IP multicast. Originally, routers simply forwarded one incoming packet to one outgoing interface. And the unicast and multicast routing protocols run by routers certainly go far beyond this definition.

Active services can provide functionality that is simply not possible end-to-end, such as splitting trust throughout a network using watercasting. It can also greatly improve the efficiency of other mechanisms, such as subcasting and NAK suppression for reliable multicast or filtering and transcoding for multimedia data. Bhattacharjee *et al.* claimed that active networks can be true to end-to-end principles because such functions work best in the network where they have access to information not accessible to end-systems [Bhattacharjee97b]. Where should the balance between this increased functionality and end-to-end benefits lie?

The considerations outlined in proposals for Generic Router Assist functionality [Cain00] are a useful base for this discussion. These include:

- Services should assist, not replace, end-to-end protocols.
- Services should not allow uploadable code or programming language functionality.

- Services should be simple. They should not require excessive processing or substantial or long-lived state.

On a scale from Deering's end-to-end Internet to the active network programmed by code in every packet, GRA and the systems we have designed lie far closer to the former. Our distributed filtering protocol meets these criteria even better than the reliable multicast protocols GRA was initially designed to support. Our distributed fingerprinting protocol is slightly more complex, but requires only a very small amount of resources above that of GRA. We believe that we can even encapsulate our work through a slight redefinition to Deering's router requirement: a router should forward an incoming packet to *zero* or more outgoing interfaces. This maintains the simplicity that allows fast and reliable operation of the router's core functionality, whilst enabling services that are difficult or impossible to provide without network assistance. Our TCP protocol also demonstrates that network information can be usefully made available to end systems, reducing the benefit described by Bhattacharjee of running in-network services.

A further distinction has been made between services that run *in* or *on* the network. Web caches and SMTP servers, for example, run on end hosts and are explicitly contacted by clients. Because of the low latency requirements of real-time communication, transcoding servers must run as directly in the path of the data flow as possible. This means that they are likely to be positioned in the network transparently to the end user. This makes the service more difficult to authenticate, and requires that security state be transferred between servers in different locations as a mobile user moves between them. It also reduces network reliability, as an extra point of failure is introduced that is difficult for an end-user to diagnose. The flexibility of the Internet in routing around failure points – its original *raison d'etre* – is compromised if a flow must travel through one of a small number of processing points.

The best way forward may be to watch the evolution of active services and their effect on the flexibility, scalability and robustness of the Internet, and make a practical rather than philosophical judgement. Reed *et al.* concluded that exceptions to the end-to-end philosophy should be considered on an individual basis, but are exceedingly rare: "few examples have turned up in 20 years of experience with systems like the Internet" [Reed98a]. Hopefully they would consider our systems worthy of inclusion in this category!

8.2 Mixing politics and engineering

Information security remains one the most controversial of scientific fields. Political requirements have been absolutely central to its direction. The drive by governments to restrict the public knowledge and use of cryptography for the last 25 years has been more constant than the science of the subject itself. Without the determination of Whit Diffie in the unclassified invention of public-key cryptography, the non-governmental world may have waited decades before benefiting from GCHQ's earlier discovery of non-secret encryption [Ellis97, Diffie99]. If NSA had succeeded in intimidating Rivest, Shamir and Adleman, particularly in aiming to take control of academic research in the area [Bamford83], Ft. Meade and Cheltenham may have remained the sole repositories of information on the subject. And if Phil Zimmerman had not packaged these obscure algorithms into a convenient tool, and fought imprisonment for many years as a result, many researchers – including the author of this thesis – may never have become interested in what otherwise would have remained a dry mathematical pursuit. The problematic combination of active networks and end-to-end security would be irrelevant if we all simply used link-layer security implemented in black boxes supplied by government agencies, as organisations such as GCHQ and NSA would have dearly loved [Banisar97, Diffie97, Levy01, Nielson96].

It is fascinating to contrast the behaviour of long-established telecommunications standards bodies such as ETSI with the IETF when wiretapping facilities in protocols have been requested by intelligence and law enforcement agencies. The former secretly made protocols

such as GSM insecure at every step of their operation [Green99, Biryukov00]. The latter consulted widely with its open membership and rejected deliberate standard insecurity as inappropriate for global protocols [IAB96, IAB00]. It seems that Internet engineers that have had the benefit of end-to-end security are far less willing to give it up at the behest of government agencies than telecommunications engineers who have always operated in a heavily-regulated environment with far greater control of their own networks.

This new assertiveness may be a problem for any organisation that tries to deploy active services that require access to plaintext. A typical comment made during the IETF's debate on this issue came from Richard Payne: "It seems that almost all of the people in favor of building in interception capabilities work for telco's and switch manufacturers. Without calling them morally obtuse, they may lack perspective on the situation. They may think that my freedom, privacy, and security for their paycheque is a reasonable trade: I don't" [Payne99].

This would be a fruitful area for economists and political scientists to study examples of reverse regulatory capture and public choice theory. Interestingly, the IETF's stance now appears to be influencing even ETSI: the latest TIPHON draft on lawful interception of IP simply states that "ETSI exists to establish and publish communications standards – the 'T' says no more and no less. ETSI therefore has no responsibility for the delivery of IP-interception standards that are beyond its ability to influence or produce." [Cadzow01a]

8.3 The future of active networks

In the short and medium term, it seems that the use of active services can only increase. The widespread delivery of multimedia over today's Internet will only be feasible if the traffic can be actively shaped to the availability of bandwidth whilst retaining the transmission efficiencies of multicast.

In the longer term, it is unclear how the rapid evolution of physical-layer technologies will affect the global availability of bandwidth. The use of Dense Wave Division Multiplexing (DWDM) over the ever-growing forest of fibre being laid by telecommunications companies will provide connectivity at Tb/s speed over much of the Internet's cores. Network traffic has previously always expanded to fill any new capacity, but it is difficult to imagine even pervasive ubiquitous computing devices and photo-realistic shared virtual environments consuming such enormous amounts of bandwidth. It has been speculated that the speed of light rather than capacity will be the constraint on future communications speed [Kelly00].

Two of our ciphertext-compatible active services perform subtractive filtering on secure streams. This is most useful for constrained bandwidth links. If round-trip time rather than congestion will be the limiting factor, these are unlikely to be a problem in 10 years on core networks. But although wireless links will also improve dramatically in capacity during that period, they have a lower theoretical total capacity and many potential new users [O'Reilly00]. Higher core bandwidth and less resource-intensive multicast protocols may mean that transcoding becomes unnecessary: sources can simply send streams containing different media formats to different groups. But base stations at least are likely to provide active services for the foreseeable future due to the extremely high variation in bandwidth available to each client at any given time. As we have described in chapter four, base stations present a completely different threat environment to the services running on your home workstation envisioned by many active network designers.

Such services may be standardised by telecommunications groups rather than the IETF. These groups are already busy attempting to retro-fit switched circuit network features into the Internet. The IETF may also come under more sustained political pressure as its standards become more and more prevalent. Active service designers must therefore be aware that the interests of government may take precedence over that of users as their service reaches the deployment stage. As Clark observed, active services are certainly useful for imposing functionality such as wiretapping upon network users [Clarke00].

The decreased cell sizes required to support higher bandwidth for more users will further emphasise the handoff problem with wireless links. Any active service targeted at mobile users will need to use small and soft amounts of state and anticipate handoffs if service is to be maintained for a user rapidly crossing between different cells.

Vastly increased bandwidth and connectivity may shift the balance away from active services back toward the reliability engendered by the use of end-to-end principles. Even long-established "in-network" services such as SMTP servers and Web caches may turn out to cause more problems than benefits. If both ends of a mail connection are almost certain to be online within a small window of time after a message is sent, why not send the message direct rather than use an intermediate store-and-forward server? That allows a secure direct delivery connection to be set up using ephemeral keys, reducing the vulnerabilities introduced by long-lived encryption keys [Brown00a] as well as removing the maintenance costs of that server. Distributed mailing lists remove the reliability problems of dependence on a central mail exploder to maintain list availability [Brown97]. Direct Web page retrieval removes complexity and the privacy problems introduced by a point that has knowledge of all pages visited by a giver user [Brown00a].

While active services can provide attractive increases in efficiency for specific applications, there is a resulting cost in reduced flexibility and reliability for all future system designers [Odlyzko99]. It is unfortunate that this incentive is so skewed in favour of individual application designers and against the overall utility of the Internet. The classic economic response to such externalities is to shift their burden back to their originator: perhaps governments or ISPs should impose an active service tax!

But splitting trust throughout the network may remain a compelling driver of active services. Our watercasting and distributed filtering protocols are two examples of such services. Anonymising HTTP, SMTP and IP proxies such as Crowds [Reiter98], mixes [Chaum81] and onion routing [Reed98b] are further instances. This is an active area of both research and commercialisation as consumers demand enhanced privacy and security from network services and content companies attempt to maintain control over the distribution of their intellectual property.

8.4 Future work

We are continuing the development of several of the protocols described in this thesis and also considering new directions for research prompted by this work.

8.4.1 Distributed packet filtering

We are now ready to perform large-scale testing of our algorithm against hostile attack. Our tests so far have been limited to twelve attackers on one switched Ethernet. We want to examine the performance of the system against hundreds of attackers distributed over a wide area network. We hope to involve the wider security community to gain access to the network connectivity and number of machines required to test on this scale.

8.4.2 Improving TCP performance over lossy links

We are implementing our protocol in the kernel of FreeBSD 4.2 in order to experimentally verify its performance. A modified IP stack at the base station labels a percentage of outgoing packets with the buffer size available for its recipient. The recipient uses this information to update its estimate of the buffer space available at the base station in one round trip time, and adjusts its TCP windows accordingly. We plan to concentrate on the performance of data transfers over LAN and WAN links from fixed to mobile hosts, as is the typical case with Web browsing and viewing a conference. This will be compared against standard and `snoop`-assisted TCP flows. We will look in detail at the effect of varying parameters such as the base station's buffer size and the buffer predictor smoothing weight on sub-component

measurements like the sender's congestion window and efficacy of buffer size prediction. We will also look at the effects of different handoff rates given varying buffer size and utilisation.

8.4.3 Watercasting

We are now planning simulations to model the performance of our method. Factors such as the size of transmission groups and complexity of filtering algorithms will have large effects on the performance of the system, and we intend to use our simulations to fine tune these parameters. We are also investigating optimisations such as increasing capacity near the multicast root for given sizes of receiver sets and the depth and breadth of the tree. We are comparing the performance of different watermarking algorithms, particularly to check for any artifacts created in watercasted data. Finally, we intend to build small test networks and distribute audiovisual and other data through them to experimentally verify our scheme and determine its implementation complexity, using these results to further develop the protocol.

The central task of watercasting is to provide *evidence*. Computer security systems often claim to reduce or obviate the need for legal solutions to a problem by removing it through technical means. Our design instead aims to provide an audit trail through which the illegal distributors of a given piece of data can be traced and prosecuted. The strength of the evidence provided by watercasting is crucial to the ability to mount a successful trial, particularly if no other evidence is available. We are working with legal researchers to determine how to best fine tune our scheme to meet this aim. We are also investigating the requirements of content providers: 'fair use' provisions in copyright laws, for example, may reduce their need for the tracing of very short clips.

Our design criteria are slightly less robust than those of, for example, the International Federation for the Phonographic Industry (IFPI), who required a watermark that could not be removed or altered "without sufficient degradation of the sound quality as to render it unusable" [IFPI97]. Our definition of unusable is not unlistenable or even unsellable, but simply perceptually intrusive enough to justify paying for the original rather than a pirated copy. We plan to use tools developed for measuring subjective audio quality [Watson98] to assess the impact of removing the watermarks we develop.

8.4.4 Trusted execution environments

Active service designers may attempt to follow the path taken by many "agent" software researchers to secure sensitive information as it is processed. Current techniques use obfuscation of code and data to hide the purpose of object code and its data from the environment it is run in [Collberg00].

There are very serious business and technical obstacles to overcome in this approach. The owner of the execution environment is unlikely to allow code that may act against their best interests to run. Why should an e-commerce site, for example, allow an agent trying to bid them down to the lowest price to browse their catalogue? Technically, there are very difficult problems in defeating an attacker with the ability to access all memory used by a process, alter its execution path and run it as many times as required with arbitrary inputs.

As a generic security problem, securing a system against an attacker with physical access has always been extremely difficult. The US Department of Defense has spent decades trying to secure nuclear warheads against theft and unauthorised detonation. Their threat model is that a successful attack should require the same effort and capability as building the weapon itself [Bellovin98]. Just one research group at Cambridge University have not yet found "any technology, or combination of technologies… which can make a smartcard resistant to penetration by a skilled and determined attacker." [Anderson01 p.291]. And the only cryptoprocessor certified to that level by the US government, IBM's low-powered 4758, costs $2000. This does not bode well for attempts to design secure active service nodes that must be reasonably cheap and located in many positions around the Internet.

Nodes with access to plaintext will anyway most likely be required to have a law enforcement access interface. Even apart from the threats outlined in chapter four, secure system design is not yet at the point where it can provide third-party access for one set of people that is absolutely secure against everyone else [Abelson97, Gladman98]. Nodes with such interfaces will therefore be able to provide only relatively poor levels of security in the medium term, so further work in this area may be a dead end as far as active services are concerned – although the research will continue to be driven by the content industries' dream of trusted content players.

This makes it particularly important that active services take advantage of cryptographic data protection rather than attempt to remove it. The multiple paranoia of mix systems that redundantly-encrypt data so that no one node has access to plaintext seems a more sensible base for further active service research. Now that large-scale systems based on mixes are being deployed, experimental investigation into the effect on security and performance of varying these encryption parameters along with other variables such as route weights would be an extremely useful next step in their development.

So far, a small class of functions have been discovered that can be efficiently executed in encrypted form. Sander and Tschudin [Sander98] showed that polynomials and rational functions can be encrypted and supplied by Alice to Bob, who applies that function to some input x and returns the result to Alice where it is decrypted. Bob has no information on the operation of the underlying function. A rigorous proof of the security and generalisability of these functions remains to be found.

More promising for active services are functions that can operate on encrypted data. Chaum's blind signatures were among the first of such functions [Chaum82]. They allow Bob to RSA-sign a "blinded" piece of information supplied by Alice, who can then "unblind" the data without invalidating the signature. Chaum suggested multiplication by a random integer $r^e \bmod n$ (where e,n is Bob's public key) as a blinding function – allowing Alice to unblind the signed data by dividing the result by $r \bmod n$. This could provide the basis, for example, for an active service that provided an audit trail for data as it moves around a network.

Proxy cryptography is a particularly promising candidate for future use in active services. It allows information encrypted with one public key K_1 to be re-encrypted to another public key K_2 without first decrypting the original ciphertext [Blaze97]. Private proxy functions require K_1 to be created with the assistance of the holder of K_2's related private key; public proxy functions do not require that assistance.

These functions would allow an active service to receive information encrypted with its own public key and then to re-encrypt that information to a set of recipients without requiring access to the plaintext of that information. This could simplify key management in a multicast security protocol: the information could be distributed in a hierarchical tree where each branch point manages the group membership one level lower in the hierarchy. This would be particularly useful in combination with a protocol such as watercasting where each recipient receives slightly different information; it would prevent one group member with access to the physical distribution network eavesdropping on and decrypting the information received by another member.

Proxy functions are currently far too slow and compute-intensive to run in active services. They could be useful in mailing lists, removing the confidentiality if not availability problems of centralised mailing list servers [Brown97]. But the functions are currently at a very early stage of development; future research could usefully investigate more lightweight functions that may be suitable for inclusion in active services.

8.4.5 Preventing observation attacks

It is well known that attackers can obtain useful information by making external observations on a system as it cryptographically processes a piece of data. Power and timing analysis allow secrets such as keys to be obtained from many smartcards by measuring the amount of power

used by the card [Kocher99] or the time it takes to perform each of a series of related tasks [Kocher96]. Even the private keys of Web servers running SSL are vulnerable to these timing attacks.

These results would be extremely helpful in developing attacks on even highly secure active service nodes that required access to the plaintext of data being processed. This further research will no doubt be carried out by open groups eager to embarrass manufacturers and closed groups keen to exploit the secrets they can capture from such nodes. And most of the efficient techniques known to reduce the effectiveness of these attacks are covered by patent protection [Shamir97, Graunke98].

This would be a useful area of continued research for those wanting to protect other types of secure systems. But timing attacks are difficult to defend against efficiently. Ideally an active service should not be in a position to even accidentally leak information about underlying plaintext – by not having access to it.

9 References

[Abelson97] H. Abelson, R. Anderson, S. Bellovin, J. Benaloh, M. Blaze, W. Diffie, J. Gilmore, P. Neumann, R. Rivest, J. Shiller and B. Schneier. The risks of key recovery, key escrow, and trusted third party encryption. *Written testimony to US Senate Judiciary Committee*, available at http://www.counterpane.com/key-escrow.pdf, May 1997.

[Allman99] M. Allman, V. Paxson and W. Stevens. TCP congestion control. *RFC 2581*, April 1999.

[Amir95] E. Amir, S. McCanne, and H. Zhang. An application level video gateway. *Proc. ACM Multimedia '95*, November 1995.

[Amir98a] E. Amir, S. McCanne and R. Katz. An Active Service Framework and its Application to Real-time Multimedia Transcoding. *Proc. ACM SIGCOMM '98*, September 1998.

[Amir98b] An Agent-based Approach to Real-time Multimedia Transmission over Heterogeneous Environments. PhD thesis, University of California at Berkeley, 1998.

[Anderson94] Ross Anderson. Liability and Computer Security: Nine Principles. *Proc. European Symposium on Research in Computer Security*, Brighton, November 1994.

[Anderson97] R.J. Anderson and C. Manifavas. Chameleon – A New Kind of Stream Cipher. *Proc. Fourth Workshop on Fast Software Encryption*, pp. 107-113, January 1997.

[Anderson01] R.J. Anderson. Security Engineering. *Chicester: John Wiley and Sons*, February 2001.

[AP01] Associated Press. Police go after Belgian Napster users. Available from http://www.politechbot.com/p-01737.html, 15 February 2001.

[Back98] A. Back and I. Brown. Reducing vulnerability to private key compromise. Available from http://www.cs.ucl.ac.uk/staff/I.Brown/pfs2.html, March 1998.

[Bakre94] A. Bakre and B.R. Badrinath. I-TCP: Indirect TCP for mobile hosts. *Technical Report DCS-TR-314*, Rutgers University, October 1994.

[Balakrishnan95] H. Balakrishnan, S. Seshan and R. H. Katz. Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks. *ACM Wireless Networks*, 1(4), December 1995.

[Balakrishnan97] H. Balakrishnan, V.N. Padmanabhan, S. Seshan and R. H. Katz. A Comparison of Mechanisms for Improving TCP Performance over Wireless Links. *IEEE/ACM Transactions on Networking*, 5(6) 756-769, December 1997.

[Ballardie95] A. Ballardie and J. Crowcroft. Multicast-Specific Security Threats and Counter-Measures. *The Internet Society Symposium on Network and Distributed System Security*, San Diego, California, February 1995.

[Ballardie96] A. Ballardie. Scalable Multicast Key Distribution. *RFC 1949*, May 1996.

[Bamford83] J. Bamford. The Puzzle Palace. *New York: Penguin*, 1983.

[Banisar97] Dave Banisar and Bruce Schneier. The Electronic Privacy Papers. *New York: Wiley*, October 1997.

[Banisar00] David Banisar and Wayne Madsen. Cryptography and Liberty 2000: An International Survey of Encryption Policy. Washington, DC: EPIC, March 2000 [http://www2.epic.org/reports/crypto2000/overview.html#Heading10].

[Barlow94] J. P. Barlow. The economy of ideas. *Wired* 2(3) pp. 85, March 1994.

[Beith01] Alan Beith MP. Intelligence Agencies debate. *Hansard* col. 1150, 29 March 2001.

[Bell99] A. E. Bell. The dynamic digital disk. *IEEE Spectrum*, 36(10) pp. 28—35, October

[Bellovin98] S.M. Bellovin. Permissive Action Links. Available at http://www.research.att.com/~smb/nsam-160/pal.html, 1999.

[Bellovin99] S.M. Bellovin. Transport-Friendly ESP. Available at http://www.research.att.com/~smb/talks/tfesp-ndss/index.htm, February 1999.

[Bellovin00] S. M. Bellovin and M. Leech. ICMP Traceback Messages. *IETF draft*, March 2000.

[Benjamin00] Simon Benjamin. Single Photons "on Demand". *Science* 290 (5500) p.2273, December 2000.

[Best80] R. M. Best. Preventing Software Piracy with Crypto-Microprocessors. *Proc. IEEE Spring COMPCON '80*, San Francisco, February 1980.

[Bhattacharjee97a] Samrat Bhattacharjee, Kenneth L. Calvert and Ellen W. Zegura. An Architecture for Active Networking. *Proc. High Performance Networking*, White Plains, NY, April 1997.

[Bhattacharjee97b] Samrat Bhattacharjee, Kenneth L. Calvert, and Ellen W. Zegura. Active Networking and the End-to-End Argument. *Proc. International Conference on Network Protocols*, Atlanta, October 1997.

[Biryukov00] Alex Biryukov, Adi Shamir and David Wagner. Real Time Cryptanalysis of A5/1 on a PC. *Proc. Fast Software Encryption*, New York City, April 2000.

[Blake98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss. An Architecture for Differentiated Services. *RFC 2475*, December 1998.

[Blaze95] M. Blaze and S. M. Bellovin. Session-Layer Encryption. *Proc. USENIX Security Workshop*, June 1995.

[Blaze97] M. Blaze and M. Strauss. Atomic Proxy Cryptography. *Proc. EuroCrypt '97*, May 1997.

[Blom01] R. Blom, E. Carrara, D. McGrew, M. Naslund, K. Norrman, and D. Oran. The Secure Real Time Transport Protocol. *IETF draft*, February 2001.

[Bluetooth99] The Bluetooth Special Interest Group. Specification of the Bluetooth System 0.9. Available from http://www.bluetooth.com/link/spec/1f4709x.pdf, May 1999.

[BNA01] Bureau of National Affairs. French Human Rights Group Sues ISPs Over Failure to Censor US-Based Hate Site. *E-Commerce Law Daily*, 20 June 2001.

[Bohm00] Nicholas Bohm, Ian Brown and Brian Gladman. Electronic commerce: who carries the risk of fraud? *Journal of Information, Law and Technology*, October 2000.

[Bolot97] J.-C. Bolot and A. Vega-García. The Case for FEC-Based Error Control for Packet Audio in the Internet. *ACM Multimedia Systems*, 1997.

[Borland01] John Borland. File-trading pressure mounts on ISPs. CNET News.com. Available at http://news.cnet.com/news/0-1004-200-6674297.html, July 2001.

[Bouch00] Anna Bouch, M. Angela Sasse and Hermann de Meer. Of packets and people: a user-centered approach to quality of service. *Proc. 8th International Workshop on Quality of Service*, Pittsburgh, June 2000.

[Bowden00] Caspar Bowden. Certificated and Overlapping Warrants, S.15.3 "safeguards" and domestic mass-surveillance. FIPR briefing note, available at http://www.fipr.org/rip/CertificatedAndOverlapping.htm, June 2000.

[Braden89] R. Braden (Ed.). Requirements for Internet Hosts – Communication Layers. RFC 1122, October 1989.

[Brassard98] G. Brassard and C Crépeau. A Bibliography of Quantum Cryptography. Available from http://www.cs.mcgill.ca/~crepeau/CRYPTO/Biblio-QC.html, July 1998.

[Bridis00] Ted Bridis. Judge Allows Delivery by E-Mail. *Associated Press*, available from http://www.politechbot.com/cyberpatrol/ap.032400.txt, 24 March 2000.

[Briscoe99] Bob Briscoe and Ian Fairman. Nark: Receiver-based Multicast Non-repudiation and Key Management. *ACM Conference on Electronic Commerce*, Denver, Colorado, November 1999.

[Brown97] Ian Brown and Ian Grigg. Distributed Mailing Lists. *Work-in-progress*, available from http://www.cs.ucl.ac.uk/staff/I.Brown/dml/dml.html, November 1997.

[Brown99a] Ian Brown, Colin Perkins and Jon Crowcroft. Watercasting: Distributed Watermarking of Multicast Media. *Proc. Networked Group Communication '99*, Pisa, Lecture Notes in Computer Science 1736 pp.286-300, November 1999.

[Brown99b] Ian Brown and C. R. Snow. A proxy approach to e-mail security. *Software - Practice and Experience*, 29(12) 1049-1060, October 1999.

[Brown00a] Ian Brown and Gus Hosein. Serve Yourself... Shifting Power Away from the Brothers. *Proc. ACM Computers, Freedom and Privacy 2000*, Toronto, April 2000.

[Brown00b] Ian Brown, Simon Davies and Gus Hosein (ed). The Economic Impact of the Regulation of Investigatory Powers Bill. *British Chambers of Commerce report*, June 2000.

[Brown01] Ian Brown. Securing emergency prioritized traffic. *ITU-T study group 16 workshop*, Santa Barbara, September 2001.

[Burke00] L. Burke. Teen Hacker's Home Raided. *Wired News* [http://www.wired.com/news/business/0,1367,33889,00.html], 25 January 2000.

[Butler98] W. T. Buttler et al. Free-space quantum-cryptography over 1 km. *Four Corners Section Meeting*, Albuquerque, New Mexico, April 1998.

[Caceres99] R. Cáceres, N. G. Duffield, J. Horowitz, D. Towsley and T. Bu. Multicast-Based Inference of Network-Internal Characteristics: Accuracy of Packet Loss Estimation. *Proc. IEEE Infocom*, New York, March 1999.

[Cadzow01a] Scott Cadzow. Issues on IP Interception. ETSI Draft Technical Report 101 944, March 2001.

[Cadzow01b] Scott Cadzow. Guide to system architectures for the Lawful Interception of telecommunications traffic. ETSI Draft Technical Report 201 xxx, April 2001.

[Cain99] Brad Cain, Steve Deering and Ajit Thyagarajan. Internet Group Management Protocol, Version 3. *IETF work in progress*, November 1999.

[Cain00] Brad Cain, Tony Speakman and Don Towsley. Generic Router Assist (GRA) Building Block Motivation and Architecture. *IETF work in progress*, November 1999.

[Callas98] J. Callas et al. OpenPGP Message Format. *RFC 2440*, November 1998.

[Campbell93] Duncan Campbell. Dispatches: The Hill. Channel 4 Television (UK), 6 October 1993.

[Campbell99] Duncan Campbell. Development of surveillance technology and risk of abuse of economic information. Report of the European Parliament Scientific and Technical Options Assessment Panel, April 1999 [http://www.gn.apc.org/duncan/stoa.htm].

[Canetti99] R. Canetti et al. An architecture for secure Internet multicast. *IRTF work in progress*, February 1999.

[Carle97] G. Carle and E. Biersack. A Survey of Error Recovery Techniques for IP-Based Audio-Visual Multicast Applications. *IEEE Network*, November/December 1997.

[Carroll98] Ken Carroll. Document Retention/Destruction Policies. Miller & Martin Advisory, available from http://www.millermartin.com/pubs/adv12_98_docret.htm, September 1998.

[CCITT88] Consultative Committee on International Telegraphy and Telephony: X.509: The Directory – Authentication Framework, 1988.

[Cerf74] V. Cerf and R. Kahn. A Protocol for Packet Network Interconnection. *IEEE Transactions on Communications*, 22, 637-648, May 1974.

[CERT01] CERT Advisory CA-2001-19: "Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL. July 2001.

[Chaum81] D. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2) 84-88, February 1981.

[Chaum82] D. Chaum. Blind signatures for untraceable payments. *Proc. Advances in Cryptology – Crypto'82*, pp. 199—203, Santa Barbara, 1982.

[Chaskar99] H. M. Chaskar, T. V. Lakshman and U. Madhow. TCP over wireless with link level error control: analysis and design methodology. *IEEE/ACM Transactions on Networking*, 7(5) 605—615, October 1999.

[Clarke00] David D. Clarke and Marjory Blumenthal. Rethinking the Design of the Internet: The End-to-end Arguments vs. the Brave New World. *Proc. 28th Telecommunications Policy Research Conference*, Alexandria, Virginia, September 2000.

[CoE00] Council of Europe. Draft Convention on Cyber-crime, April 2000 [http://conventions.coe.int/treaty/en/projets/cybercrime.htm].

[Collberg00] Christian Collberg and Clark Thomborson. Watermarking, Tamper-Proofing, and Obfuscation – Tools for Software Protection. Proc. DIMACS Workshop on Management of Digital Intellectual Property. Rutgers University, NJ, USA, April 2000.

[Conta98] A. Conta and Steve Deering. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. *RFC 2463*, December 1998.

[Cope90] B. J. B. Cope. Biometric systems for access control. *Electrotechnology*, 71-74, April/May 1990.

[Crowcroft01] Jon Crowcroft. TCP friendliness. Post to *Reliable Multicast Transport* working group, rmt@listserv.lbl.gov, April 2001.

[Daugman94] J. G. Daugman. Biometric personal identification system based on iris analysis. *US patent 5,291,560*, March 1994.

[Davies94] J. Davies. Personal identification devices and access control systems. *US patent 5,608,387*, May 1994.

[Davies98] Simon Davies. Travellers face jail if they refuse to let Customs scan their laptops. *The Daily Telegraph*, 16 September 1998 [http://www.sightings.com/political/laptops.htm].

[Davies00] Simon Davies and Ian Brown. Expert defence witness report for *Reg. vs. Connell*, Redbridge Youth Court, 19—20 October 2000.

[Day83] J. D. Day and H. Zimmerman. The OSI Reference Model. *Proceedings of the IEEE*, 71, 1334-1340, December 1983.

[Deering89] Steve Deering. Host Extensions for IP Multicasting. *STD 5/RFC 1112*, August 1989.

[Deering91] Steve Deering. Multicast Routing in a Datagram Internetwork. *PhD thesis*, Stanford University, December 1991.

[Deering99] Steve Deering. The IP Hourglass. *Networked Group Communication '99*, Pisa, November 1999.

[Denning96] D. E. Denning and P. F. MacDoran. Location-Based Authentication: Grounding Cyberspace for Better Security. *Computer Fraud and Security*, February 1996

[Dierks99] T. Dierks and C. Allen. The TLS Protocol Version 1.0. *RFC 2246*, January 1999.

[Diffie76] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6) 644—654, November 1976.

[Diffie92] W. Diffie, P. van Oorschot and M. Wiener. Authentication and Authenticated Key Exchanges. *Designs, Codes and Cryptography*, 2 107—125, 1992.

[Diffie97] W. Diffie and S. Landau. Privacy on the Line: The Politics of Wiretapping and Encryption. *Cambridge: MIT Press*, 1997.

[Diffie99] W. Diffie. Non-secret encryption and public key cryptography. Available from http://www.cs.ucl.ac.uk/staff/I.Brown/nse-pkc.html, April 1999.

[DN97] VW will pay out $1.1 billion to settle GM lawsuit. Detroit News, available at http://www.detroitnews.com/96/biz/9701/10/01100113.htm, 10 January 1997.

[Dobbertin96] H. Dobbertin. The Status of MD5 After a Recent Attack. *RSA Labs' CryptoBytes*, 2(2), available from http://www.rsa.com/rsalabs/pubs/cryptobytes.html, Summer 1996.

[DoD85] Department of Defense Trusted Computer System Book. *DOD 5200.28-STD*, December 1985

[Dwork96] C. Dwork, J. Lotspiech and M. Naor. Digital Signets: Self Enforcing Protection of Digital Information. *Proc. 28th Annual ACM Symposium on the Theory of Computing*, Philadelphia, pp. 489, May 1996.

[Eastlake97] D. E. Eastlake. Secure Domain Name System Dynamic Update. *RFC 2137*, April 1997.

[Eastlake99] D. E. Eastlake. Domain Name System Security Extensions. *RFC 2535*, March 1999.

[Eck85] W. van Eck. Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk? *Computers and Security* 4:269-285, December 1985.

[Elfving00] Judge William J. Elfving. DVD CCA v. McLaughlin, Bunner et al. Case CV 786804, *US S. Court Calif.* January 2000.

[Elgamal84] T. Elgamal. A public-key cryptosystem and a signature scheme based on discrete logarithms. *Advances in Cryptology: Proc. Crypto '84*, 10—18, 1984.

[Ellis97] J. H. Ellis. The Story of Non-Secret Encryption. *CESG Report*. Available at: http://www.cesg.gov.uk/about/nsecret/ellis.htm, 1997.

[Ellison00] Carl Ellison and Bruce Schneier. Risks of PKI: E-Commerce. *Communications of the ACM*, 43(2) pp.152, February 2000.

[EP97] European Parliament. Resolution on the Commission communication on illegal and harmful content on the Internet. April 1997.

[Erramilli87] A. Erramilli and R. P. Singh. A reliable and efficient multicast protocol for broadband broadcast networks. *Proc. ACM SIGCOMM '87*, 343—352, August 1987.

[ETSI99] European Telecommunications Standards Institute. Handover interface for the lawful interception of telecommunications traffic. ETSI Standard 201 671, July 1999.

[FBI97] Federal Bureau of Investigation Annual Financial Statement Fiscal Year 1996 Audit Report 97-29A, August 1997.

[Floyd91] Sally Floyd. Connections with Multiple Congested Gateways in Packet-Switched Networks Part 1: One-way Traffic. *Computer Communication Review*, 21(5) pp.30—47, October 1991.

[Floyd95] Sally Floyd et al. A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing. *Proc. ACM SIGCOMM'95*, Cambridge, Massachusetts, August 1995.

[Frackman00] R. J. Frackman *et al*. Complaint for contributory and vicarious copyright infringement, violations of California civil code section 980(a)(2), and unfair competition to the United States District Court for the Northern District of California. Available at http://www.riaa.com/PDF/Napster_Complaint.pdf, March 2000.

[Fusaro93] Roberta Fusaro. Cases highlight need for e-mail policies. *ComputerWorld*, available from http://www.computerworld.com/home/print.nsf/(frames)/9810056D2A?OpenDocument&˜f 10 May 1998.

[Gladman98] Brian Gladman. Key recovery – meeting the needs of users or key escrow in disguise? *EPIC Cryptography and Privacy Conference*, Washington DC, June 1998.

[Gladman99] Brian Gladman, Nicholas Bohm and Ian Brown. Could New Chip Privacy and Security Measures Tie Users' Hands? *Comment to Intel Corp.*, available from http://www.cs.ucl.ac.uk/staff/I.Brown/chip-sec.htm, July 1999.

[Gladman00] Brian Gladman. The Regulation of Investigatory Powers Bill – The Provisions for Government Access to Keys. *Foundation for Information Policy Research report*, February 2000 [http://www.fipr.org/rip/RIPGAKBG.pdf].

[Glass00] S. Glass, T. Hiller, S. Jacobs and C. Perkins. Mobile IP Authentication, Authorization, and Accounting Requirements. *IETF work in progress*, January 2000.

[Goodin98] Dan Goodin and Jeff Pelline. Smoking gun in Microsoft memos? CNET News.com, May 1998 [http://news.cnet.com/news/0-1003-200-329400.html].

[Graunke98] Gary L. Graunke and David W**.** Aucsmith. Method and apparatus for hiding crytographic keys utilizing autocorrelation timing encoding and computation. *US patent 6,041,122*, February 1998.

[Green99] Lucky Green. More NSAKEY musings. In B. Schneier (ed.), *Crypto-gram*, September 1999.

[Handley97] Mark Handley. An examination of Mbone performance. *USC/ISI research report ISI/RR-97-450*, April 1997.

[Handley98a] Mark Handley and Van Jacobson. SDP: Session Description Protocol. *RFC 2327*, April 1998.

[Handley98b] Mark Handley and D. Thaler. Multicast-Scope Zone Announcement Protocol. *IETF draft*, October 1998.

[Handley99a] Mark Handley et al. SAP: Session Announcement Protocol, *IETF work in progress*, June 1999.

[Handley99b] Mark Handley et al. SIP: Session Initiation Protocol. *RFC 2453*, March 1999.

[Handley00] Mark Handley, Colin Perkins and Edmund Whelan. Session Announcement Protocol. *RFC 2974*, October 2000.

[Hanzo94] L. Hanzo and D. Chandler. Performance of the MPEG Video Codec. *Southampton University Research Journal*, 1994.

[Hardjano98a] T. Hardjono, B. Cain and N. Doraswamy. A Framework for Group Key Management for Multicast Security. *IRTF work in progress*, July 1998.

[Hardjano98b] T. Hardjono, N. Doraswamy and B. Cain. An Architecture for Conference-Support using Secured Multicast. *Proc. IFIP 8th International Conference on High-Performance Networking*, September 1998.

[Harkins98] D. Harkins and D. Carrel. The Internet Key Exchange. *RFC 2409*, November 1998.

[Hartley99] D. Hartley, UK Defence Evaluation and Research Agency. Personal communication, May 1999.

[Henderson01] Tristan Henderson, Jon Crowcroft and Saleem Bhatti. Congestion pricing: paying your way in communication networks. IEEE Internet Computing, 2001.

[Hinsch96] E. Hinsch et al. The Secure Conferencing User Agent: A Tool to Provide Secure Conferencing with Mbone Multimedia Conferencing Applications. *Proc. IDMS '96*, Berlin, March 96.

[Holbrook99] H. Holbrook and D. Cheriton. IP Multicast Channels: EXPRESS Support for Large-scale Singlesource Applications. *Proc. ACM SIGCOMM'99*, Harvard, September 1999.

[Hu00] Jim Hu and Evan Henson. Yahoo auction case may reveal borders of cyberspace. *CNET News.com*, available from http://news.cnet.com/news/0-1005-200-2495751.html, August 2000.

[IAB96] The Internet Architecture Board and the Internet Engineering Steering Group. IAB and IESG statement on cryptographic technology and the Internet. *RFC 1984*, August 1996.

[IAB00] The Internet Architecture Board and the Internet Engineering Steering Group. IETF policy on wiretapping. *RFC 2804*, May 2000.

[IEEE97] Institute of Electrical and Electronics Engineers. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. *IEEE Std. 802.11*, November 1997.

[IFPI97] International Federation of the Phonographic Industry. Request for proposals – Embedded signalling systems issue 1.0. June 1997.

[Ireland00] Ireland Electronic Commerce Act 2000, 6 April 2000 [http://www.entemp.ie/ecd/ecb12000.pdf].

[ITU98] International Telecommunication Union: H.235, Security and encryption for H series (H.323 and other H.245 based) multimedia terminals, February 1998.

[Jacobson88] V. Jacobson. Congestion avoidance and control. *Proc. ACM SIGCOMM '88*, August 1988.

[Jacobson92] V. Jacobson, R. Braden and D. Borman. TCP Extensions for High Performance. *RFC 1323*, May 1992.

[Kahn67] David Kahn. The Codebreakers: The Story of Secret Writing. New York: The Macmillan Company, 1967.

[Kane99] M. Kane. Divx dies – DVD the big winner. ZDNet News, available at http://www.zdnet.com/zdnn/stories/news/0,4586,2276806,00.html, June 1999.

[Katz97] D. Katz. IP Router Alert Option. *RFC 2213*, February 1997.

[Kelly00] Frank P. Kelly. Models for a self-managed Internet. *Philosophical Transactions of the Royal Society, Mathematical, Physical and Engineering Sciences A* 358(1773) 2159—2358, August 2000.

[Kent98a] S. Kent and R. Atkinson. IP Authentication Header. *RFC 2402*, November 1998.

[Kent98b] S. Kent and R. Atkinson. IP Encapsulating Security Payload. *RFC 2406*, November 1998.

[Khare98] R. Khare. Upgrading to TLS Within HTTP/1.1. *IETF work in progress*, November 1998.

[Kirstein98] P. T. Kirstein et al.: Accessing Mbone Sessions over Point-to-Point Connections. Technical report , Department of Computer Science, University College London, Gower Street, London WC1E 6BT, UK, 1998.

[Knuth69] Donald E. Knuth. *The Art of Computer Programming*. Reading: Addison-Wesley, 1969.

[Kocher96] Paul C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. *Advances in Cryptology – CRYPTO 96*, Santa Barbara, August 1996.

[Kocher98] Paul C. Kocher, Joshua Jaffe and Benjamin Jun. Differential Power Analysis. *Advances in Cryptology – CRYPTO 98*, 1998.

[Kojo97] M. Kojo et al. An efficient transport service for slow wireless telephone links. *IEEE Journal on Selected Areas of Communication*, 15(7) 1337-1348, September 1997.

[Kömmerling99] O. Kömmerling and M. G. Kuhn. Design Principles for Tamper-Resistant Smartcard Processors. *Proc. USENIX Workshop on Smartcard Technology*, May 1999.

[Krawczyk97] H. Krawczyk, M. Bellare and R. Canetti. HMAC: Keyed-Hashing for Message Authentication. *RFC 2104*, February 1997.

[Kuhn98] M. G. Kuhn and R. J. Anderson. Soft Tempest: Hidden Data Transmission Using Electromagnetic Emanations. *Proc. Second Workshop on Information Hiding*, April 1998.

[LeGall91] D. Le Gall. MPEG: a video compression standard for multimedia applications. *Communications of the ACM*, 34(4):46—58, April 1991.

[Lessig99] Lawrence Lessig. Code and other laws of cyberspace. *New York: Basic Books*, December 1999.

[Levine98] B. Levine and J. J. Garcia-Luna-Aceves. A Comparison of Reliable Multicast Protocols. *ACM Multimedia Systems Journal*, August 1998.

[Levine99] B. Levine, S. Paul and J. J. Garcia-Luna-Aceves. Organizing Multicast Receivers Deterministically by Packet-Loss Correlation. Preprint, University of California, Santa Cruz.

[Levy01] Steven Levy. Crypto: How the Code Rebels Beat the Government – Saving Privacy in the Digital Age. Viking Press, January 2001.

[Liou91] M. L. Liou. Overview of the p*64 kbit/s video coding standard. *Communications of the ACM*, 34(4):59—63, April 1991.

[Loscocco98] P. Loscocco *et al*. The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments. *Proc. 21st National Information Systems Security Conference*, October 1998.

[Love01] James Love. What you should know about The Hague Conference on Private International Law's Proposed Convention on Jurisdiction and Foreign Judgments in Civil and Commercial Matters. Consumer Project on Technology Report, June 2001.

[Luby01] M. Luby, J. Gemmell, L.Vicisano, L. Rizzo and J. Crowcroft. Asynchronous Layered Coding: A massively scalable reliably content delivery protocol. *Internet draft*, July 2001.

[Ludwig99] R. Ludwig, B. Rathonyi, A. Konrad, K. Oden and A. D. Joseph. Multi-layer Tracing of TCP over a Reliable Wireless Link, *Proc. ACM SIGMETRICS 1999*.

[Macavinta99] Courtney Macavinta. Yahoo message board suit continues. CNET News.com, March 1, 1999 [http://news.cnet.com/news/0-1005-200-339276.html].

[Mankin91] A. Mankin and K. Ramakrishnan (Ed). Gateway Congestion Control Survey. *RFC 1254*, August 1991.

[Mankin01] Allison Mankin, Dan Massey, Chien-Long Wu, S. Felix Wu and Lixia Zhang. On Design and Evaluation of "Intention-Driven" ICMP Traceback. 2001.

[Maughan98] D. Maughhan, et al. Internet Security Association and Key Management Protocol. *RFC 2408*, November 1998.

[McAuliffe01] Megan McAuliffee. Excite@Home snoops on user downloads. *ZDNet News*, 22 August 2001.

[McCanne96a] S. McCanne, V. Jacobson and M. Vetterli. Receiver-driven Layered Multicast. *Proc. ACM SIGCOMM '96*, Stanford, CA, August 1996.

[McCanne96b] S. McCanne. Scalable Compression and Transmission of Internet Multicast Video. *Ph.D. thesis*, University of California Berkeley, UCB/CSD-96-928, December 1996.

[McCullagh96] Declan McCullagh. Helsingius shuts down anon.penet.fi server in Finland. *Politech*, August 1996 [http://www.itu.reading.ac.uk/misc/mailing_lists/rre/00000167.htm].

[McCullagh00] Adrian McCullagh and William Caelli. Non-Repudiation in the Digital Environment. *First Monday* 5(8), August 2000.

[Meyer96] G. Meyer. The PPP Encryption Control Protocol (ECP). *RFC 1968*, June 1996.

[Meyer98] D. Meyer. Administratively Scoped IP Multicast. *RFC 2365*, July 1998.

[Moon98] S. Moon, J. Kurose, P. Skelly and D. Towsley. Correlation of Packet Delay and Loss in the Internet. Technical Report 98-11, Department of Computer Science, University of Massachusetts, Amherst, MA 01003, USA.

[Moore01] David Moore. The Spread of the Code-Red Worm. CAIDA analysis, August 2001. Available at http://www.caida.org/analysis/security/code-red/

[MSN00] MSN. Madonna Home Page. Available at http://www.msn.co.uk/madonna/, November 2000.

[Murray98] W. H. Murray. UK Customs Check for laptop porn. Posts to `talk.politics.crypto`, August 1998 [http://privacy.nb.ca/cryptography/archives/cryptography/html/1998-08/0111.html].

[Naor88] M. Naor and B. Pinkas. Threshold Traitor Tracing. *Advances in Cryptology – CRYPTO '88*, Lecture Notes in Computer Science 403, August 1988.

[nCipher00] nCipher Corporation. Secure Execution Engine White Paper. Available at http://www.ncipher.com/products/rscs/white_papers/SEE_White_Paper.pdf, 2000.

[Nielson96] K. W. Nielson and J. Thorel. Interview with David Herson. *Engineering Weekly*, October 1996.

[NIST88] National Institute of Standards and Technology. Data Encryption Standard. Federal Information Processing Standards Publication 46-1, January 1988.

[NIST91] National Institute of Standards and Technology. Proposed Federal Information Processing Standards for Digital Signature Standard (DSS). Federal Register 56(169) 42980—42982, August 1991.

[NIST94] National Institute of Standards and Technology. Standard Security Label for Information Transfer. Federal Information Processing Standards Publication 188, September 1994.

[NIST01] National Institute of Standards and Technology. Draft Federal Information Processing Standard. Announcing the Advanced Encryption Standard. February 2001.

[Nuttall98] Chris Nuttall. UK Customs check for laptop porn. *BBC News Online,* 13 August 1998 [http://news.bbc.co.uk/hi/english/sci/tech/newsid_150000/150465.stm].

[Odlyzko99] Andrew M. Odlyzko. The stupid network: Essential yet unattainable. *ACM netWorker,* 3(4) pp.36—37, December 1999.

[Odlyzko01] Andrew M. Odlyzko. Internet growth: Myth and reality, use and abuse. J*ournal of Computer Resource Management*, April 2001.

[OECD97] OECD Guidelines for Cryptography Policy, December 1997 [http://www.oecd.org/dsti/sti/it/secur/prod/crypto2.htm].

[Oppliger97] R. Oppliger. Internet Security: Firewalls and Beyond. *Communications of the ACM,* 40(5), 92-102, May 1997.

[O'Reilly00] John O'Reilly. Broadband wireless systems and networks. *Philosophical Transactions of the Royal Society, Mathematical, Physical and Engineering Sciences A* 358(1773) 2159—2358, August 2000.

[Parnes96] P. Parnes et al. mTunnel: a multicast tunneling system with a user based Quality-of-Service model, *Proc. European Workshop on Interactive Multimedia Systems and Telecommunication Services*, September 1997.

[Payne99] Richard Payne. Some considerations. *IETF Raven Working Group,* available at http://www.ietf.org/mail-archive/working-groups/raven/current/msg00141.html, October 1999.

[Perkins96] C. Perkins. IP Mobility Support. *RFC 2002*, October 1996.

[Perkins97] C. Perkins et al. RTP Payload for Redundant Audio Data. *RFC 2198*, September 1997.

[Perkins98] C. Perkins, O. Hodson and V. Hardman. A Survey of Packet Loss Recovery Techniques for Streaming Audio. *IEEE Network*, pp. 40—49, September/October

[Perkins99a] C. Perkins and D. B. Johnson. Route Optimization in Mobile IP. *IETF work in progress*, January 1999.

[Perkins99b] C. Perkins et al. A Message Bus for Conferencing Systems, *IETF work in progress*, August 1999.

[Petitcolas98] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn. Attacks on copyright marking systems. *Proc. Second Workshop on Information Hiding*, April 1998.

[Pike98] John Pike. Intelligence Agency Budget and Personnel. Federation of American Scientists, July 1998 [http://www.fas.org/irp/agency/budget1.htm].

[Pingali97] S. Pingali, D. Towsley and J. Kurose. A comparison of sender-initiated and receiver-initiated reliable mlticast protocols. *IEEE Journal on Selected Areas of Communication*, 15, April 1997.

[Postel81] J. Postel (Ed.). Transmission Control Protocol Specification. *RFC 793*, September 1981.

[Postel97] J. Postel and J. Reynolds. Instructions to RFC Authors. RFC 2223, October 1997.

[Rahnema93] M. Rahnema. Overview of the GSM System and Protocol Architecture. *IEEE Commucation Magazine,* 31(4) pp.92—100, April 1993.

[Rakoff00] Jed Rakoff. Universal Music Group vs. MP3.com. United States District Court for the Southern District of New York, 6 September 2000.

[Ramsdell99] B. Ramsdell. S/MIME Version 3 Message Specification. *RFC 633*, June 1999.

[Rasmusson95] L. Rasmusson. About the data format of WB. http://www.it.kth.se/~d90-lra/wb-proto.html, December 1995.

[Ratnasamy99] S. Ratnasamy and S. McCanne. Inference of Multicast Routing Trees and Bottleneck Bandwidths using End-to-end Measurements. *Proc. IEEE Infocom '99*, New York, March 1999.

[Redl95] S. M. Redl, M. K. Weber, and M. W. Oliphant. An Introduction to GSM. Boston: Artech House, 1995.

[Reed98a] David P. Reed, Jerome H. Saltzer and David D. Clark. Comment on Active Networking and End-to-End Arguments. *IEEE Network*, 12 (3) 69-71, May 1998.

[Reed98b] M. G. Reed, P. F. Syverson and D. M. Goldschlag. Anonymous Connections and Onion Routing. IEEE Journal on Selected Areas of Communication: Special Issue Copyright and Privacy Protection, May 1998.

[Reiter98] M. Reiter and A. D Rubin. Crowds: anonymity for Web transactions. *ACM Transactions on Information and System Security*, 1(1) 66-92, November 1998.

[Rescorla98] E. Rescorla. HTTP Over TLS. *IETF work in progress*, September 1998.

[Rivest78] R. Rivest, A. Shamir and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2) 120—126, February 1978.

[Rivest92] R. Rivest. The MD5 Message Digest Algorithm. *RFC 1321*, April 1992.

[Rizzo97] L. Rizzo. An Embedded Network Simulator to Support Network Protocols' Development. *Proc. Tools '97*, St. Malo, June 1997.

[Rizzo00] L. Rizzo. pgmcc: a TCP-friendly single-rate multicast congestion control scheme. *Proc. SIGCOMM 2000*, Stockholm, August 2000.

[Rohatgi99] P. Rohatgi. A Hybrid Signature Scheme for Multicast Source Authentication. *IRTF work in progress*, June 1999.

[Romano00] Bill Romano. Ex-Cisco employee convicted in theft of critical source code. *Mercury News*, available at http://www0.mercurycenter.com/svtech/news/indepth/docs/cisco050300.htm, 3 May 2000.

[Rufford00] Nicholas Rufford. MI5 builds new centre to read e-mails on the net. *The Sunday Times*, 30 April 2000 [http://www.thetimes.co.uk/news/pages/sti/2000/04/30/stinwenws01034.html].

[Saltzer84] Jerome H. Saltzer, David P. Reed and David D. Clark. End-to-end arguments in system design. *ACM Transactions in Computer Systems* 2(4) pp. 277-288, November, 1984.

[Samuelson96] Pamela Samuelson. A Prohibition Law Glides over the Internet. International UNESCO Symposium on the Effects of New Technology on Cultural Information, Transmission and Dissemination, the Protection of Authors´Rights and Other Holders of Rights, Madrid, March 1996.

[Samuelson98] Pamela Samuelson. Does information really have to be licensed? *Communications of the ACM*, 41(9) 15—20, September 1998.

[Samuelson99] Pamela Samuelson. Intellectual Property and Contract Law for the Information Age. *California Law Review* 87(1), January 1999.

[Sander98] Tomas Sander. Protecting Mobile Code. *Proc. International Symposium on Software Reliability Engineering*, Albuquerque, November 1998.

[Santos98] J. Santos and D. Wetherall. Increasing Effective Link Bandwidth by Suppressing Replicated Data. *Proc. Usenix Annual Technical. Conference*, June 1998.

[Savage99] Stefan Savage, Neal Cardwell, David Wetherall and Tom Anderson. TCP Congestion Control with a Misbehaving Receiver. *ACM Computer Communications Review*, 29(5) pp. 71-78, October 1999.

[Savage00] Stefan Savage, David Wetherall, Anna Karlin and Tom Anderson. Practical Network Support for IP Traceback. *Proc. ACM SIGCOMM*, Stockholm, August 2000.

[Schroeder01] Mary Schroeder. A&M v. Napster. *United States Ninth Circuit Court of Appeals*, October 2000.

[Schneier96] Bruce Schneier. *Applied Cryptography* (2nd ed.). New York: John Wiley and Sons, 1996.

[Schulzrinne96] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson. RTP: A Transport Protocol for Real-time Applications. *RFC 1889*, January 1996.

[Schulzrinne98] H. Schulzrinne et al. Real Time Streaming Protocol (RTSP). *RFC 2326*, April 1998.

[Sergienko96] Greg S. Sergienko. Self Incrimination and Cryptographic Keys. *The Richmond Journal of Law and Technology*, February 1996 [http://www.richmond.edu/jolt/v2i1/sergienko.html].

[Seshan97] S. Seshan, H. Balakrishnan and R. H. Katz . Handoffs in Cellular Wireless Networks: The Daedalus Implementation and Experience. *Kluwer International Journal on Wireless Personal Communications*, January 1997.

[Shamir97] Adi Shamir. Method and apparatus for protecting public key schemes from timing and fault attacks. *US patent 5,991,415*, May 1997.

[Shane96] Scott Shane. Mixing business with spying; secret information is passed routinely to US. *Baltimore Sun*, 1 November 1996.

[Shepherd95] S. J. Shepherd. Continuous Authentication by Analysis of Keyboard Typing Characteristics. *European Convention on Security and Detection*, May 1995.

[Sherwin00] David Sherwin. Fraud – the unmanaged risk. *Financial Crime Review* 1(1) pp.67—69, Fall 2000.

[Shostack95] A. Shostack. An Overview of SHTTP. Available from http://www.homeport.org/~adam/shttp.html, May 1995.

[Simmons97] Peter L. Simmons. Records Retention Policies: When Is It Safe To Destroy Documents. *Fried, Frank, Harris, Shriver & Jacobson client memos*, available from http://www.ffhsj.com/firmpage/cmemos/131280.htm, January 1997.

[Simpson94] W. Simpson. The Point-to-Point Protocol (PPP). *RFC 1661*, July 1994.

[Simpson96] W. Simpson. PPP Challenge Handshake Authentication Protocol (CHAP). *RFC 1994*, August 1996.

[Snoeren01] Alex C. Snoeren, Craig Partridge, Luis A. Sanchez, Christine E. Jones, Fabrice Tchakountio, Stephen T. Kent and W. Timothy Strayer. Hash-Based IP Traceback. *Proc. ACM SIGCOMM*, San Diego, August 2001.

[Sommer98] Peter Sommer. Digital Footprints: Assessing Computer Evidence. *Criminal Law Review* (Special Edition), pp.61-78, December 1998.

[Song01] Dawn Xiaodong Song and Adrian Perrig. Advanved and Authenticated Marking Schemes for IP Traceback. *Proc. IEEE Infocom*, Alaska, April 2001.

[Speakman99] Tony Speakman, Nidhi Bhaskar, Richard Edmonstone, Dino Farinacci, Steven Lin, Alex Tweedly, Lorenzo Vicisano and Jim Gemmell. PGM Reliable Transport Protocol Specification. *IETF work in progress*, 1999.

[Spring00] Neil T. Spring and David Wetherall. A protocol-independent technique for eliminating redundant network traffic. *Proc. SIGCOMM 2000*, Stockholm, August 2000.

[Stevenson99] F. Stevenson. More on CSS. Post to cryptography@c2.net [http://www.mail-archive.com/cryptography@c2.net/msg02322.html], November 1999.

[Swan98] A. Swan et al. Layered Transmission and Caching for the Session Directory Service, *Proc. ACM Multimedia '98*, Bristol, UK, September 1998.

[Tallo01] Ivan Tallo. Report of the Committee on Legal Affairs and Human Rights, Parliamentary Assembly, Council of Europe on the Draft Convention on Cybercrime, April 2001.

[Tanenbaum98] Andrew S. Tanenbaum. *Computer Networks* (3rd edition). Simons and Schuster, May 1998.

[Tennenhouse97] D. L. Tennenhouse et al. A survey of active network research. *IEEE Communications Magazine*, 80-86, January 1997.

[Thayer98] R. Thayer, N. Doraswamy and R. Glenn. IP Security Document Roadmap. *RFC 2411*, November 1998.

[Thompson97] K. Thompson, G. J. Miller and R. Wilder. Wide-area Internet traffic patterns and characteristics. *IEEE Network* 11(6) pp. 10—23, November/December 1997.

[Todd00] Bennett Todd. Distributed Denial of Service Attacks. *Linux Security*, February 2000. Available at http://www.linuxsecurity.com/resource_files/intrusion_detection/ddos-faq.html#Overview

[Urban96] Mark Urban. *UK Eyes Alpha*. London: Faber and Faber, 1996, p.235.

[US06] Hale v. Henkel, 201 U.S. 43, 74—75, March 1906.

[US81] Executive Order 12333 – United States intelligence activities. 46 FR 59941, 3 CFR, December 1981 [http://www.nara.gov/fedreg/eos/e12333.html].

[US90] U.S. Army Corps of Engineers. Electromagnetic Pulse (EMP) and Tempest Protection for Facilities. *Engineer Pamphlet EP 110-3-2*, December 1990.

[Vicisano98] L. Vicisano, L. Rizzo and J. Crowcroft. TCP-like congestion control for layered multicast data transfer. *Proc. IEEE Infocom*, San Francisco, March 1998.

[Wallner98] D. Wallner et al. Key Management for Multicast: Issues and Architectures. *IETF work in progress*, September 1998.

[WAP98] The Wireless Application Protocol Forum. WAP 1.0 Specification Suite. April 1998.

[WAP01] The Wireless Application Protocol Forum . WAP TLS Profile and Tunneling Specification. April 2001.

[Watson98] A. Watson and M. A. Sasse. Measuring Perceived Quality of Speech and Video in Multimedia Conferencing Applications. *Proc. ACM Multimedia '98*, Bristol, England, pp. 55-60, September 1998.

[Wetherall99] David Wetherall. Active network vision and reality: lessons from a capsule-based system. *Operating Systems Review* 34(5) pp.64–79, December 1999.

[Windrem00] Robert Windrem. U.S. steps up commercial spying. MSNBC News, 7 May 2000 [http://msnbc.com/news/403435.asp?cp1=1].

[Woolsey00] John Woolsey. Foreign Press Center, Washington, DC briefing, 7 March 2000 [http://cryptome.org/echelon-cia.htm].

[Yajnik96] M. Yajnik, J. Kurose and D. Towsley. Packet Loss Correlation in the Mbone Multicast Network. *Proc. IEEE Global Internet Conference*, November 1996.

[Yeadon96] Nicholas Yeadon. Quality of Service Filters for Multimedia Communications. *Ph.D. Thesis*, Lancaster University, May 1996.

[Ylonen99] T. Ylonen et al. SSH Protocol Architecture. *IETF work in progress*, February 1999.

[Young00] J. Young. Re: DVD-deCSS Court case. Post to ukcrypto@maillist.ox.ac.uk, available from http://www.chiark.greenend.org.uk/pipermail/ukcrypto/2000-February/007761.html, February 2000.

[Zenel97] B. Zenel. A Proxy Based Filtering Mechanism for the Mobile Environment. *Ph.D. thesis, Columbia University*, 1997.