WHIM: Watermarking Multicast Video with a Hierarchy of Intermediaries

Paul Judge, Mostafa Ammar
College of Computing, Georgia Institute of Technology, Atlanta, GA 30332
{judge,ammar}@cc.gatech.edu

Abstract— **Fingerprinting, watermarking content to identify the recipient, provides a good deterrence to unauthorized duplication and propagation of multimedia streams. This is straightforward in a unicast environment; however, in a multicast environment, inserting a fingerprint at the source does not provide any security since many receivers will share a common fingerprint. A simple solution would be to fingerprint the data for each user at the source and unicast the different streams. We aim to achieve a more scalable solution while maintaining and even increasing the level of security. To achieve this, we have developed WHIM, a scalable system that allows multicast content to be marked with distinct information for distinct receivers securely. This system introduces two new concepts: 1)generating a watermark based on the receiver's location in a tree overlaying the network and 2)inserting the watermark in the content incrementally as it traverses an overlay network. We propose and evaluate several forms of this architecture and show how it improves scalability while increasing security.**

I. Introduction

As content distribution on the Internet becomes more pervasive and the value of the content being distributed increases, the security of this data has become a main concern of content providers. Encryption is generally used to safeguard the content while it is being transmitted so that unauthorized persons can not read the stream from the network, but this offers no protection after the intended receiver receives the data. There is no protection against unauthorized duplication and propagation by the intended receiver. This additional protection can be obtained by watermarking the content. Watermarking is the embedding of some identifying information into the content in such a manner that it can not be removed by the user but it can be extracted or read by the appropriate party. Watermarks can be used for copyright protection or for identification of the receiver. Copyright protection watermarks embed some information in the data to identify the copyright holder or content provider, while receiver-identifying watermarking, commonly referred to as *fingerprinting* [1], embeds information to identify the receiver of that copy of the content. Thus, if an unauthorized copy of the content is recovered, extracting the fingerprint will show who the initial receiver was.

Problems arise when attempting to fingerprint content in a multicast environment that do not arise in copyright protection watermarking. Copyright protection watermarks are embedded in the data at the source, then the watermarked data is multicast to the group of receivers. For fingerprinting, embedding the receiver's identification as the watermark at the source will not work since all the receivers will share the same watermark. It is necessary to watermark content with unique information for distinct receivers of the same multicast stream. A simple method to achieve unique watermarks for each receiver would be to watermark the stream differently for each receiver and to unicast the watermarked streams. Of course, the inefficiency of such a scheme calls for a better solution. We aim to maintain the security of this approach while achieving scalability.

We propose WHIM, a scalable system that allows multicast content to be securely marked with distinct information for distinct receivers. This system introduces two new concepts: 1)generating a watermark based on the receiver's location in the network and 2) inserting the watermark in the content incrementally as it traverses the network. WHIM makes use of a hierarchy of intermediaries for creating and embedding the fingerprint. This allows security and scalability. The use of a hierarchy allows a new type of security by having an User ID based on the user's location in an overlay network. Security is also maintained by using proven watermarking algorithms to embed this User ID. The hierarchy leads to scalability by capitalizing on the efficiency of multicast distribution and by distributing the watermark embedding load from the source to the different intermediaries.

This paper proceeds as follows. In Section 2 we enumerate the design objectives of WHIM. Section 3 gives an overview of the WHIM architecture. Section 4 discusses the WHIM-Backbone component which is based on a hierarchy of intermediaries that provide an efficient distribution architecture that fingerprints the streaming content. Section 5 describes the WHIM-Last Hop component, a secure protocol that fingerprints and distributes content between an intermediary and a group of receivers. Section 6 examines previous work in the area. Section 7 presents an analysis and simulation results of the efficiency of WHIM, and a comparison with previous solutions. Finally, section 8 presents conclusions and discusses possible future work.

II. WHIM Objectives

The design objectives of a system to fingerprint multicast content should be security and scalability. We outline the concepts involved in achieving these goals. The features and components of the system necessary to accom-

This work supported by grants from the National Science Foundation, DoD MURI Program, and the Georgia Tech Information Security Center

plish these goals should be designed into the solution.

Security:

Robustness of the fingerprinting method:

The fingerprint is what distinguishes one user from another. This can be a particular pattern of frames or a particular pattern embedded in a frame. The method used must be robust to efforts of a user to remove this distinguishing information. There has been significant work in video watermarking see for example [2], [3], [4], [5]. A scheme extending these efforts into fingerprinting multicast content is desirable since it assures a robust fingerprinting method.

Collusion problem: Collusion is when a set of group members work together to use the set of differently watermarked streams to create a copy of the content which cannot be determined to contain the fingerprint of any of those receivers. The solution must be based on a fingerprinting scheme that is not susceptible to collusion.

Asymmetric fingerprinting: Schemes should be able to provide asymmetric fingerprinting. This allows the sender to identify the receiver of a recovered copy of data without previously knowing the fingerprinted data. Thus, the sender is not capable of distributing the data and accusing an innocent receiver. [6]

Protection Granularity: The granularity of protection is the amount of content that is needed for the protocol to be able to determine the receiver of the content. Schemes should be able to provide the smallest possible protection granularity but also be flexible so that this can be changed depending on the needs of the application.

Scalability:

Logging Requirements: Logging is necessary because once a video is recovered and the fingerprint is extracted, there must be some record of what receiver was represented by the ID recovered from the watermark at that instant in time. The storage and processing overhead of logging should be minimum.

Efficiency: The efficiency of the solution is based on the amount of data that the source must transmit and encrypt and the amount of data introduced into the network.

III. WHIM Architecture Overview

The system has two components, WHIM Backbone (WHIM-BB) and WHIM Last Hop (WHIM-LH). WHIM-BB introduces a hierarchy of intermediaries into the network and forms an overlay network between them. Figure 1 shows how the hierarchy is formed as an overlay network in the physical network. Each intermediary has a unique ID. Based on the fact that there exists a unique path between the source and each intermediary on this overlay network, we use this path to distinguish between intermediaries. Each path is identified by the IDs of the intermediaries on the path. This Path ID is embedded into the content to identify the path that it traveled. Each intermediary embeds its portion of the Path ID into the content as it forwards the content through the network. This embedding is done using modified versions of existing video watermarking algorithms. This is along the lines of the recent trend towards introducing a hierarchy of entities into the network to provide active services, such as reliable multicast [7], [8], Internet caching [9], [10], [11], multimedia proxy servers [12], and layered video multicast [13].

Each intermediary can have child intermediaries as well as a set of child receivers. We call this set of child receivers the intermediary's domain. A watermark embedded by WHIM-BB identifies the domain of a receiver. Some literature suggests that identifying the domain of the receiver or the last hop before the receiver is adequate protection [14]; however, we feel that it is necessary in many applications to identify the individual receiver. So, we propose WHIM-LH which allows intermediaries to mark the content distinctly for any children receivers that they might have. WHIM-LH forms a domain-wide secure distribution and fingerprinting system including key distribution and logging.

A central component of WHIM-LH is a secure client-side fingerprint insertion program that communicates with the intermediary for registration and to receive the decryption keys and the stream. The security of this component can be achieved by using techniques such as Mobile Cryptography [15] and Time Limited Black box Protection [16]. Clients join and register for the group at the domain level. This type of control is ideal for applications in which domains are responsible for the activity of its members. For example, a university might subscribe to a site-wide license for a broadcast then have students subscribe individually to receive it.

WHIM-LH is a building block that when merged with WHIM-BB forms a complete solution for fingerprinting multicast content distinctly for each receiver in the group. Used together, WHIM-BB and WHIM-LH allow content to be marked to pinpoint the location of the receiver in the overlay network as well as to identify the individual receiver. WHIM protects against attacks in which receivers join a group using a fake IP address or name. Even if the WHIM-LH registration fails to lead to the actual receiver, the WHIM-BB Path ID will pinpoint the responsible domain. It should be noted that either of these can be used alone as a complete fingerprinting system. WHIM-BB, alone, offers a fingerprinting system that identifies the domain of the receiver, but not the individual receiver. WHIM-LH can be used between the source and the group of receivers to fingerprint the content uniquely for each receiver. However, it lacks the scalability of the combined solution due to the lack of the distributed architecture and it does not provide any information regarding the location of the receiver.

IV. WHIM Backbone (WHIM-BB)

WHIM-BB makes use of a hierarchy of intermediaries for creating and embedding the fingerprint. The fingerprint is based on the path from the source to the intermediary. This allows security and scalability. Use of a hierarchy allows a new type of security by having the user's fingerprint based on the user's location in the network. Security is also main-

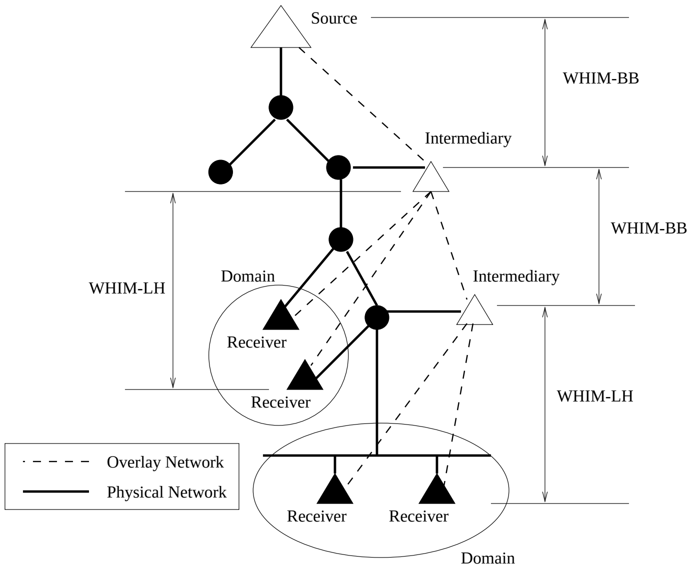

Fig. 1. The Hierarchy of Intermediaries as an Overlay Network

tained by using proven watermarking algorithms to embed this identifying information. The hierarchy leads to scalability by distributing the watermark embedding load from the source to the different intermediaries and by easing logging requirements. This section first describes the architecture of intermediaries, then discusses the distributed watermarking algorithms used by the intermediaries, and finally, discusses the logging necessary to maintain the path information.

A. Architecture

The architecture consists of a hierarchy of intermediaries positioned as endsystems in the network. Each intermediary is assigned a unique ID either manually or using some prefix labeling algorithm [17]; so to identify the intermediary, there exists a unique ID that identifies each path from the source to each intermediary. As the content traverses the network, every intermediary through which it passes concatenates its ID to the Path ID already embedded in the content.

The amount of computation required to insert the watermark is more than routers today are capable of and possibly even more than the amount of processing power proposed by advocates of active networking [18], [19]. Therefore, WHIM-BB places a hierarchy of intermediaries as endsystems in the network and forms an overlay network between them. This overlay network can be rapidly deployed and easily managed by the use of a system such as the X-bone [20]. As explained in [21], use of such a distribution architecture can help avoid many of the problems involved in using an IP multicast distribution model such as congestion control and end-to-end reliability and even increase security. This hierarchy can use application-layer multicast rather than rely on global IP multicast support while still allowing the use of IP multicast where available, especially within domains.

This idea can be extended to allow the intermediaries to be coupled with existing machines in the network that perform computation. Infrastructures in place for multimedia proxy servers [12], server replication, and caching [10], [11], [9] provide ideal locations for WHIM intermediaries to be located.

B. Distributed Watermarking Algorithms

The watermark consists of a timestamp and the concatenation of all the IDs of the intermediaries on the path and is inserted in each frame. Now, we examine how to use existing watermarking methods and distribute the computation across the intermediaries.

Example 1 The watermarking algorithm described in [2] works as follows. For each frame, a pseudo random sequence is calculated to determine the order in which the blocks will be marked. In the determined order, the blocks are discrete cosine transformed, smooth and edge detection is done, and the blocks are quantitized with Qm/Qf accordingly. For each block, the information is embedded as in the Zhao-Koch algorithm [22].

Our distributed version of this algorithm performs as follows. The source creates the pseudo random sequence in which the blocks will be watermarked, does smooth and edge detection for each block, and quantitizes with Qm/Qf. The watermark begins with a timestamp inserted by the source. It then sends the new frame and the sequence towards the receivers. As each intermediary receives the stream, it uses the sequence to determine the next blocks to watermark, adds its ID to the watermark, and sends the remainder of the sequence and new frame towards the group.

Example 2 The watermarking algorithm proposed in [5] works as follows. An adaptive scheme is used to choose the blocks to be watermarked. Smooth and edge detection is done to determine the blocks that can withstand watermarking. Also, within each block, coefficients to be used to embed the bit are chosen pseudo-randomly based on properties of the block. The information is embedded by modifying these chosen coefficients based on the Zhao-Koch algorithm.

Modified to perform in a distributed environment, the algorithm operates as follows. The source does smooth and edge detection and selects coefficients for each block. After beginning the watermark with a timestamp, the source sends the sequence of blocks to be watermarked and which coefficients are to be changed along with the stream towards the receivers. As each intermediary receives the stream, it uses the sequence to determine the next blocks to watermark and which coefficients in that block to use. The intermediary then adds its ID to the watermark and sends the rest of the block sequence and coefficient information along with the altered frame towards the group.

It should be noted that each frame containing the entire string of identifying information does not imply concentration of the watermark. It simply means that the en-

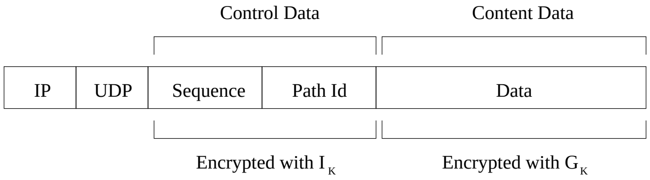

Fig. 2. Packet transmitted between Intermediaries

tire piece of identifying information is embedded into each frame. The embedding algorithm is still based on a secure watermarking algorithm which effectively hides the embedded information inside of that frame data. Therefore, there is no reduction in the level of security due to this.

If there is not a need to safeguard single frames or very short clips, selective watermarking [23] can be used to increase the performance. This involves a trade-off in the strength of the security because the length of video clip that is necessary to extract the watermark is increased. Instead of inserting the fingerprint in every frame, it can be inserted in every n-th frame. This translates into about a n-fold increase in performance with a tradeoff of n times the length of the clip that is necessary to extract the watermark. For example, with an MPEG stream, it is possible to fingerprint only I frames. If the MPEG stream has the repeating IBBPBBPBB pattern, this will reduce the computational overhead by reducing the numbers of frames that are fingerprinted by 89%.

The information exchanged by the intermediaries is encrypted with an intermediary group key, I_k, while the content data is encrypted with some session key, G_k, as shown in Figure 2. In cases in which the intermediary does not already have the compressed video data available, it will need to perform the necessary decapsulations, possibly including RTP [24], UDP, and IP, to extract the video data. Once the video data is available, the intermediary must perform the steps to locate the necessary blocks and embed the watermark. An example of this algorithm for MPEG video is shown in Figure 3.

C. Logging

In order to be able to determine the domain of the receiver from retrieved watermarks, the log must have enough information so that it can determine which nodes were represented by that Path ID at that particular instant in time. Previously, there has not been much attention to the logging aspect of such a watermarking system. We have identified it as a key requirement of the system and an important factor in the scalability of the system. While previous schemes for fingerprinting multicast video require extended periods of the fingerprinted video in order to extract enough information about the embedded fingerprint to determine the recipient, WHIM requires only one frame since the entire label is inserted in each frame. Thus, WHIM can safeguard each frame of a video. With some other schemes, if a user illegally redistributes a single image or a very short clip from a video, there is no way of determining the perpetrator. Also, our logging system requires only minimal information and uses a simple and straightforward algorithm to determine receivers.

Each intermediary sends to the logging system, the Path ID that has accumulated up to and including it. This Path ID includes the timestamp inserted by the source. Depending on the overlay management used, the intermediary might also send its IP address or some other identifying information. This includes some authentication information so that the logging system is assured that the information is being received from a legitimate intermediary. This logging information is sent to the log every time that the Path ID of the intermediary changes. Therefore updates are only sent when the overlay topology changes, not every time the underlying routing topology changes. When a watermark has been extracted and the receiver must be determined, only a simple table lookup algorithm is necessary to access this information from the log.

V. WHIM Last-Hop (WHIM-LH)

This section describes WHIM-LH, a protocol between a single intermediary and its children receivers. Whereas WHIM-BB marks the content to identify the last hop intermediary of a receiver, WHIM-LH allows a single intermediary to embed distinct User IDs for each of its children receivers. This section first explains the WHIM-LH architecture and the variations that are allowed by the different types of User IDs. Then, the different methods that are available for choosing User IDs are explained.

A. Methods of Transporting the Video Data

This architecture allows the efficiency in the network that is the motivation of multicasting while enforcing the necessary security at the endpoints, the intermediary and the client. There is significant research in the area of video watermarking, so we aim to provide a framework which will allow any watermarking algorithm to be used to fingerprint multicast streams efficiently. We introduce a secure client-side fingerprint insertion program that contains a watermarking module that can be based on any chosen watermarking algorithm. Figure 4 shows the interaction between the modules of the architecture. The intermediary distributes the fingerprinting program with a built-in decryption key, denoted as program[$K_{internal}$]. The client registers with the logging and key distribution system to join the group and receives decryption keys and possibly a User Id. The client program then receives the stream encrypted with the session key, denoted as $\{\text{stream}\}K_{play}$, from the intermediary and securely adds the watermark before making the stream available to the user. The remainder of this subsection explains the variations of this architecture depending on the type of User ID used. The *Assigned User ID* scheme has the intermediary communicate with the group using the following steps:

```
Repeat
    Extract Picture ?
    Repeat
        Extract Slice ?
        Repeat
            Extract Macroblock ?
            Repeat
                Extract Luminance Block ?
                Transform VLC To Integers
                Embed Bit By Altering Coefficients For This Block
                Convert Integers Back To VLC
            Until (Finished Inserting Bits) Or (No More Luminance Blocks In This Macroblock)
        Until (Finished Inserting Bits) Or (No More Macroblocks In This Slice)
    Until (Finished Inserting Bits) Or (No More Slices In This Picture)
Until (Finished Inserting Bits) Or (No More Pictures In This Sequence Layer)
```

Fig. 3. The bit-embedding algorithm at the intermediary.

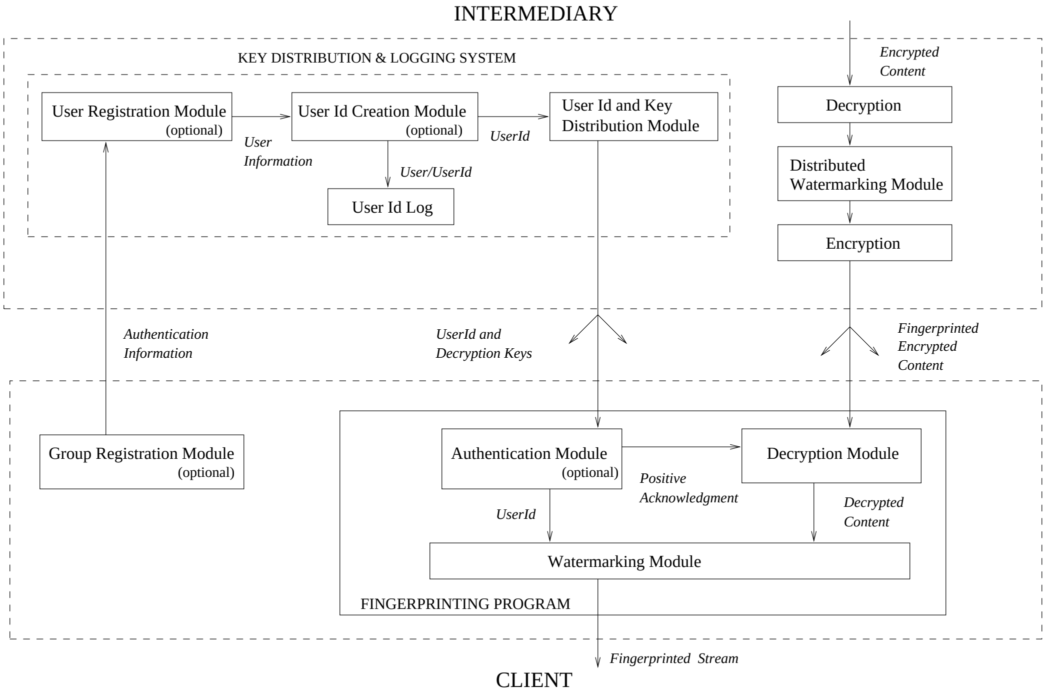

Fig. 4. WHIM-LH Architecture

```
Intermediary to Receivers:
   Multicast:  {stream}K_play
   Multicast:  program[K_internal]
   Multicast:  {{K_play,User ID1}K_internal}K_user1,
               {{K_play, User ID2}K_internal}K_user2, ...
               {{K_play, User IDn}K_internal}K_usern
```

Each User ID and key packet is encrypted with the user's public key or symmetric key shared by the logging system and the user, so the same level of security is achieved as if they were unicast. A significant portion of the traffic that is sent is the User ID information.

For applications that would benefit from the decrease in traffic that would result from not sending this information, we propose a method that allows the user to provide her own User ID information to the program. This *Local User ID* method only requires the intermediary to send the following messages to the group:

```
Intermediary to Receivers:
   Multicast:  {stream}K_play
   Multicast:  program[K_internal]
   Multicast:  {K_play}K_internal
```

The Authentication module authenticates the user and signals the decryption module. This approach is used when the logging system already has a mapping between the User ID and the actual receiver or can determine the receiver based on the User ID, such as when the User ID is derived from the public key as explained in the next subsection.

B. Methods of Choosing User ID

The User ID information that is embedded as the watermark uniquely identifies each receiver. While previous literature simply refers to the User ID as some unique identifier, perhaps randomly assigned, we propose a new technique for creating User IDs. By using cryptographic means, we compose a User ID that is more closely bound to a user than a randomly assigned User ID. As shown in the previous subsection, this also allows a more efficient distribution method. Possible methods of forming a User ID include the following ways:

- **Assigned User ID:** This simple scheme involves each user registering with the source, being authenticated, and being assigned some unique value as a User ID.
- **Public Key-based User ID:** This approach allows the User ID to be based on the public key of the receiver. This requires the user to have a public key certificate [25], a signed message from a trusted certification authority(CA) that specifies the user's name and the corresponding public key, such as a X.509 certificate [26]. The fingerprinting program must be assured that the public key used is the one assigned to this user by the CA. We suggest two methods of doing this. The fingerprinting program requests the user's public key from the CA and then uses a nonce to confirm that the user knows the corresponding private key. The second method is that the user provides the program with the public key certificate and signs it with the private key. Thus, the program can verify the public/private key pair and that it was assigned by the CA.

C. Discussion

WHIM-LH provides a framework that allows proven watermarking algorithms to be used efficiently in a multicast environment. It allows efficient rekeying, introduces a new type of secure User ID construction, has the smallest possible protection granularity, and is efficient. It also is capable of being used with selective watermarking [23] to increase its efficiency. Figure 5 shows the how the WHIM-LH architecture is combined with WHIM-BB.

We propose means of preventing the risk of the fingerprinting program being reverse engineered to reveal the decryption key or otherwise altered to disallow the desired results. There are a number of attacks that malicious users can perform against mobile agents including spying out code and data and manipulation of code and data [27]. Mobile Cryptography can be used to guard against these attacks [15]. This involves executing encrypted functions to guarantee code privacy and code integrity. Time Limited Black box Protection [16] can be used to protect the code and data of a mobile agent from being read or modified for at least some minimal time interval.

VI. Related Work

Chu, Qiao, and Nahrstedt [28] proposed a protocol to provide a different version of a multicast stream to each group member. The protocol creates two watermarked MPEG streams, assigns a unique random binary sequence to each user, and uses this sequence to arbitrate between those two watermarked streams. For the ith watermarked frame in stream $j (j = 0, 1)$, a different key KEY_i^j is used to encrypt it. Then user n is given either KEY_i^0 orKEY_i^1 depending on the random bit sequence of user n. The efficiency of this protocol is hampered by the need to watermark, encrypt, and transmit two copies of the stream and by the significant amount of key messages that the protocol transmits. However, the protocol does have a problem with the collusion issue. The ability of the protocol to detect a collusion is dependent on the length of the retrieved data stream. Even with a retrieved data stream of sufficient length, the algorithm to determine a collusion is so complex that there is not a known length of retrieved stream that can guarantee a c-collusion detection. The protection granularity of this protocol is large since it is based on the number of receivers.

Wu and Wu [23] proposed a technique which selectively encrypts and watermarks segments of an MPEG video, unicasts these and multicasts the remainder of the video. Depending on the specific selection scheme used, the chosen segments could be from 90% to less than 1% of the original video. There is a tradeoff between efficiency and security. As smaller amounts of the video are chosen for encryption and watermarking, the ability of persons outside of the group to obtain the video increases due to the proposal of not encrypting the video that is not watermarked and the ability of group members to obtain video that is not watermarked increases due to the fact that if only I frames are watermarked, then unwatermarked I-blocks found in P and B frames can provide some degree of quality video. As larger percentages of the video are chosen to be watermarked, encrypted, and unicast, the security increases, but the efficiency of the protocol begins to resemble that of the simple unicast model. Since only I frames, are watermarked, the protection granularity is each set of the I-frame pattern.

Brown, Perkins, and Crowcroft [14] recently proposed a technique that has each group member receive a slightly different version of the multicast video stream. For a multicast group with a tree of depth d, the source creates n differently watermarked copies of each packet such that $n > d$. On receiving a transmission group of packets, each router forwards all but one of the packets. The last hop router then forwards exactly one packet to the subnet with the receiver(s). The goal is that each receiver then receives a stream that consists of a unique combination of watermarked packets. The original receiver of a recovered stream can be determined by simulating the operation of various network components during the time that clip was originally transmitted. This makes the logging requirements high since the log must keep the state of the entire network from the start to the end of the transmission. The requirement that the source watermark, encrypt, and transmit n copies of the stream makes this solution inefficient. The paper does not offer a solution for having multiple receivers on the same subnet since they will have the same User ID. Also, there is no mention of the ability of receivers on a link that is not a leaf to obtain access to the set of packets that

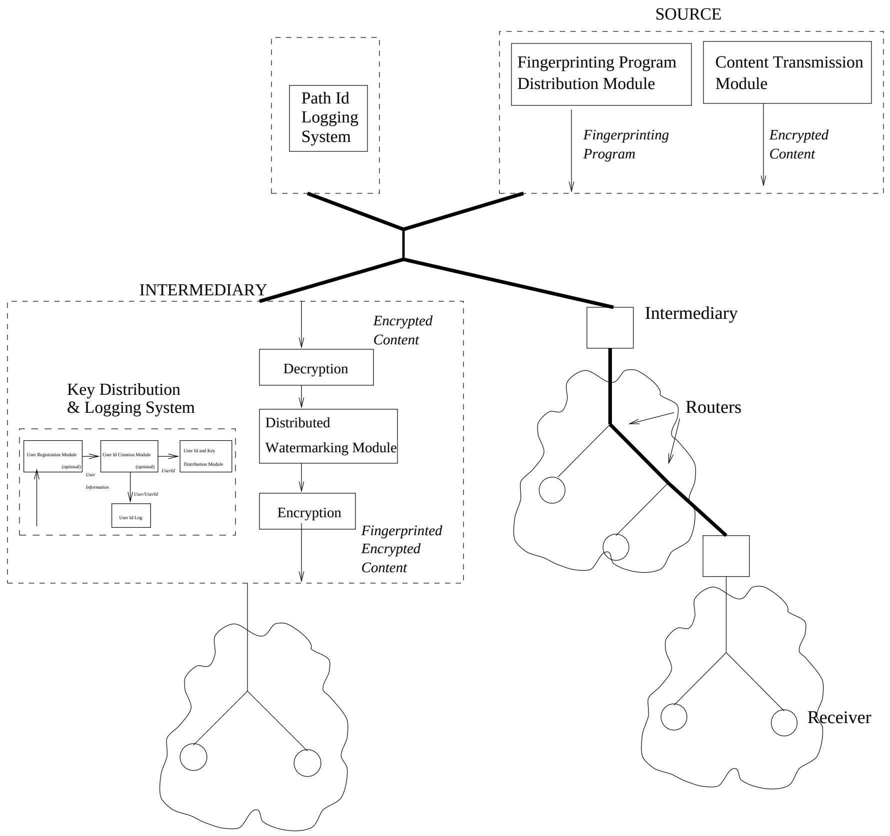

Fig. 5. WHIM

s	=	stream
nf	=	number of frames in the stream
f	=	frame
p	=	program
n	=	number of group members
ku	=	key/User ID message
cku	=	combined key/User ID messages
k	=	decryption key
uid	=	User ID
bit	=	signifies which stream the user receives
kf	=	decryption key for a particular frame

Fig. 6. Definition of Variables Used in Analysis

are traversing that link. As the length of the clip increases, the probability of being able to specify a single receiver increases. Thus, the protection granularity is large. Also, the ability to determine collusions is dependent on the length of the clip and requires extensive computation to determine what users could have possibly had access to the frames in the recovered stream.

VII. Analysis

In this section we examine the efficiency of WHIM in terms of data transmission and encryption overhead. We look at this relative to the performance of some of the other multicast watermarking schemes reviewed in the related work section. Figure 6 shows the definitions of variables used in this section.

In WHIM, the source transmits $s+p+cku$ bytes and encrypts $s+(n)(ku)$ bytes. The overhead of the scheme in [28] involves the sender transmitting $nf[2(f)+2(kf)]$ bytes, then the group leader transmits $nf[(n)(uid+bit+kf)]$ bytes. This system also has significant encryption overhead, $nf[2(f)+kf+msg]$ for the sender and $nf[(bit+kf)(n)+msg]$ for the leader. In the protocol of [14], the amount of transmitted data is increased substantially by the redundant data that is necessary. For a stream of size, s, the amount of data that is transmitted is at least $n*s$, where $n>d$ and d is the depth of the multicast tree.

We seek to analyze the performance of these schemes with two different types of group behavior, theater-style and dynamic. Theater-style involves all of the group members arriving or joining the group and leaving the group at approximately the same time, as at a movie theater. This allows all of the set up overhead to be multicast to the entire group at once. Dynamic groups involve users joining and leaving the group at anytime throughout the session and may involve members leaving and re-joining. This also involves rekeying of the group.

In order to analyze the performance for theater-style groups, we created multicast groups within transit-stub internetwork topologies using GT-ITM [29]. For each group size, the depth used in our data is based on the average depth of the 10 random multicast trees that were created. We compare the total amount of data transmitted and encrypted by the multicast source in WHIM with the schemes of [14] and [28] in Figure 7. These calculations are based

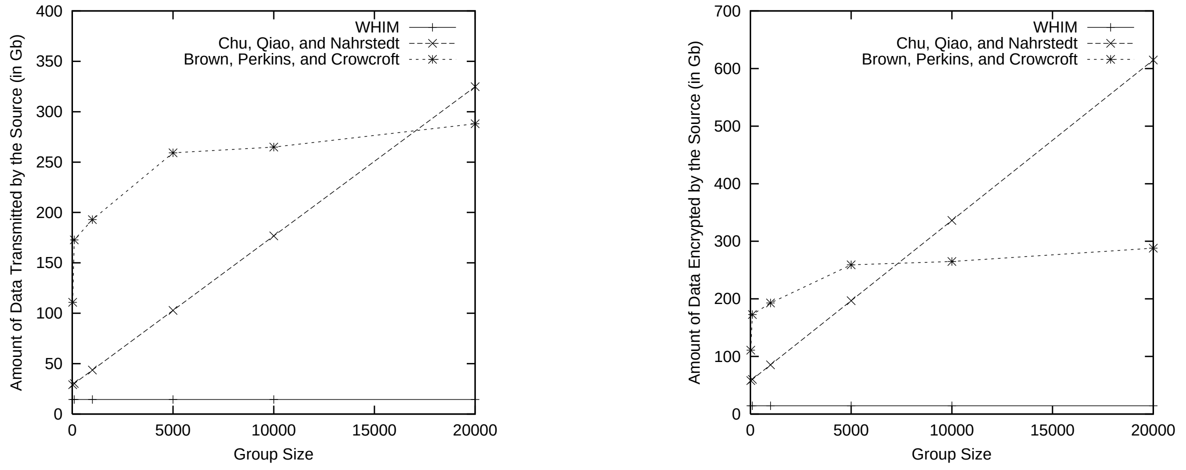

Fig. 7. Total amount of data processed by the source relative to group size, transmitted data and encrypted data

on the source multicasting a one hour session of MPEG-2 video at 4Mbps at a framerate of 30. The size of the keys in [28] and WHIM are 128 bits. In [28], the source is also the group leader. The size of the insertion program in WHIM was determined by adding the size of a common decryption program and the size of a watermarking program to be 1MB; However, the total amount of data transmitted and encrypted by the architecture is orders of magnitude above the size of the program so the accuracy of this value becomes insignificant.

For dynamic groups, we used data collected by the Mlisten tool [30] over several days for the Mbone multicast of the Space Shuttle Mission STS-80 in November 1996. This session has a duration of 13 days and has over 1600 join requests. We used these traces to simulate the performance of the fingerprinting solutions. Figure 8 shows the cumulative amount of data transmitted over the network by these schemes as the session continues and the number of receivers in the group over time.

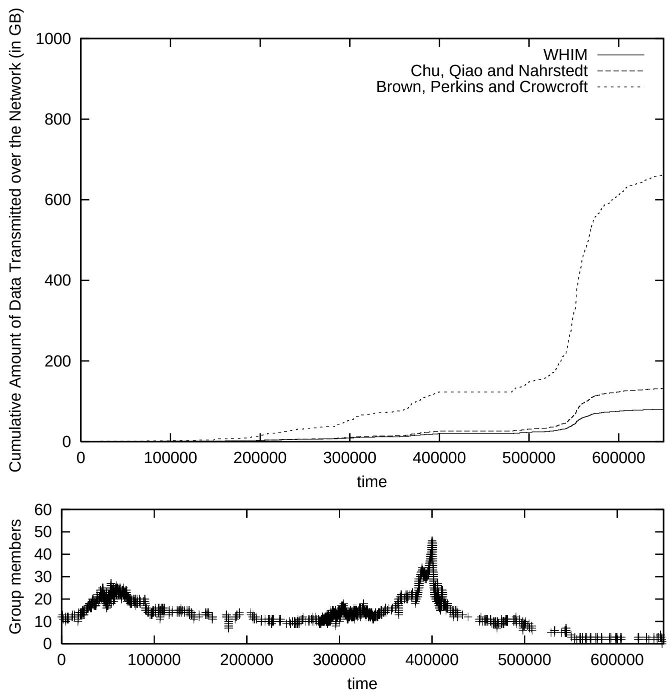

Fig. 8. Cumulative Amount of Data Transmitted over the Network through the session

VIII. Conclusions

There has been a significant amount of work geared toward developing algorithms to securely embed watermarks into multimedia content. The work presented in this paper complements these efforts by providing an architecture that allows these algorithms to be used in multicast multimedia. We have presented two architectures, WHIM-Backbone, a hierarchy of intermediaries that provides an efficient distribution architecture that fingerprints the streaming content, and WHIM-Last Hop, a secure client/server protocol that fingerprints and distributes content between a single entity and a group of receivers, which form WHIM. Our analysis shows the efficiency gains of WHIM over previous solutions.

Table I compares the trust, scalability, and resolution achieved by solutions based on the type of transmission of the video and the marking location of the data. The first column shows the simple case of marking at the source and unicasting. This achieves high trust and resolution but low scalability. The next column shows multicast video that is marked at the source. This results in high trust and scalability but low resolution. The third column shows WHIM-LH which multicasts the video and marks at the client. This achieves medium trust and scalability and high resolution. The fourth column shows WHIM-BB which multicasts the video and marks at the intermediaries. This achieves high trust and scalability and medium resolution. The last column shows WHIM which combines WHIM-LH and WHIM-BB to achieve the scalability of multicast with the trust and resolution of a unicast approach.

In addition to the architecture presented in this paper, the idea of identifying a user by his position in the network can be carried over into other applications to offer increased security and the use of a trusted hierarchy to

Transmission of Video	Unicast	Multicast			
Marking Location	Source	Source	Client (WHIM-LH)	Intermediary (WHIM-BB)	Intermediary and Client (WHIM)
Trust	High	High	Medium	High	High
Scalability	Low	High	Medium	High	High
Resolution	High	Low	High	Medium	High

TABLE I

Comparison of trust, scalability, and resolution provided by different methods of fingerprinting content to a group

provide scalable security functionality can be used in other areas including group key management, firewalls, and defending denial-of-service attacks.

References

[1] N. R. Wagner, "Fingerprinting," in *Proceedings of the 1983 Symposium on Security and Privacy*, (Oakland, California), pp. 18–22, IEEE, Apr. 1983.

[2] J. Dittmann, M. Stabenau, and R. Steinmetz, "Robust mpeg video watermarking technologies," in *Proc. Of ACM Multimedia*, 1998.

[3] L. Qiao and K. Nahrstedt, "Watermarking method for mpeg encoded video: Towards resolving rightful ownership," in *IEEE Multimedia Computing and Systems*, June 1998.

[4] F. Hartung and B. Girod, "Digital watermarking of raw and compressed video," in *European EOS/SPIE Symposium on Advanced Imaging and Network Technologies*, (Berlin, Germany), October 1996.

[5] M. J. Holliman, N. D. Memon, B.-L. Yeo, and M. M. Yeung, "Adaptive public watermarking of dct-based compressed image," in *Storage and Retrieval for Image and Video Databases (SPIE)*, pp. 284–295, 1998.

[6] B. Pfitzmann and M. Waidner, "Asymmetric fingerprinting for larger collusions," in *4th ACM Conference on Computer and Communications Security*, pp. 151–160, 1997.

[7] J. C. Lin and S. Paul, "RMTP: a reliable multicast transport protocol," in *IEEE Infocom*, (San Fransisco, California), Mar. 1996.

[8] S. Paul, K. K. Sabnani, J. C.-H. Lin, and S. Bhattacharyya, "Reliable multicast transport protocol (RMTP)," *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 407–421, Apr. 1997.

[9] L. Fan, P. Cao, J. Almeida, and A. Broder, "Summary cache: A scalable wide-area web cache sharing protocol," in *ACM SIGCOMM*, vol. 28, pp. 254–265, Sept. 1998.

[10] A. Chankhunthod, P. Danzig, C. Neerdaels, M. F. Schwartz, and K. J. Worrell, "A hierarchical internet object cache," in *USENIX 1996 Annual Technical Conference*, (San Diego, California), Jan. 1996.

[11] D. Karger, E. Lehman, T. Leighton, M. Levine, D. Lewin, and R. Panigrahy, "Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web," in *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, (El Paso, Texas), pp. 654–663, 4–6 May 1997.

[12] S. Sen, J. Rexford, and D. Towsley, "Proxy prefix caching for multimedia streams," in *IEEE Infocom*, (New York), Mar. 1999.

[13] X. Li, S. Paul, and M. Ammar, "Layered video multicast with retransmissions (LVMR): evaluation of hierarchical rate control," in *IEEE Infocom*, (San Francisco, California), p. 1062, March/April 1998.

[14] I. Brown, C. Perkins, and J. Crowcroft, "Watercasting: distributed watermarking of multicast media," in *Networked Group Communication '99*, (Pisa, Italy), pp. 286–300, Nov. 1999.

[15] T. Sander and C. F. Tschudin, "Protecting mobile agents against malicious hosts," in *Mobile Agents and Security* (G. Vigna, ed.), Springer-Verlag, 1998.

[16] F. Hohl, "Time limited blackbox security: Protecting mobile agents from malicious hosts," in *Mobile Agents and Security* (G. Vigna, ed.), Springer-Verlag, 1998.

[17] E. Bakker and R. T. J van Leeuwen, "Prefix routing schemes in dynamic networks," *Computer Networks and ISDN Systems*, vol. 26, pp. 403–421, 1993.

[18] K. L. Calvert, S. Bhattacharjee, E. Zegura, and J. Sterbenz, "Directions in active networks," *IEEE Communications Magazine*, vol. 36, pp. 72–78, Oct. 1998.

[19] D. L. Tennenhouse, J. M. W. Smith, D. Sincoskie, D. J. Wetherall, and J. G. M. Minden, "A survey of active network research," *IEEE Communications Magazine*, vol. 35, pp. –, Jan. 1997.

[20] J. Touch and S. Hotz, "The X-bone," in *Third Global Internet Mini-Conference in conjunction with Globecom*, (Sydney, Australia), Nov 1998.

[21] P. Francis, "Yallcast: Extending the internet multicast architecture," unrefereed report, NTT Information Sharing Platform Laboratories, September 1999.

[22] E. Koch and J. Zhao, "Towards robust and hidden image copyright labeling," in *IEEE Workshop on Nonlinear Signal and Image Processing*, 1995.

[23] T.Wu and S. Wu, "Selective encryption and watermarking of mpeg video," tech. rep., North Carolina State University.

[24] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a transport protocol for real-time applications," Request for Comments (Proposed Standard) 1889, Internet Engineering Task Force, Jan. 1996.

[25] L. Kohnfelder, "Towards a practical public-key cryptosystem," Master's thesis, M.I.T., May 1978.

[26] R. Housley, W. Ford, W. Polk, and D. Solo, "Internet X.509 public key infrastructure certificate and CRL profile," Request for Comments (Proposed Standard) 2459, Internet Engineering Task Force, Jan. 1999.

[27] F. Hohl, "A model of attacks of malicious hosts against mobile agents," in *4th Workshop on Mobile Object Systems: Secure Internet Mobile Computations*, 1998.

[28] H. hua Chu, L. Qiao, and K. Nahrstedt, "A secure multicast protocol with copyright protection," in *Proceedings of IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology*, January 1999.

[29] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *IEEE Infocom*, (San Fransisco, California), Mar. 1996.

[30] K. C. Almeroth and M. H. Ammar, "Multicast group behavior in the internet's multicast backbone (MBone)," *IEEE Communications Magazine*, vol. 35, June 1997.