

# MANAGING PRIVACY DATA IN PERVASIVE CAMERA NETWORKS

*Sen-ching S. Cheung, Jithendra K. Paruchuri*

Electrical & Computer Engineering  
University of Kentucky  
Lexington, KY - 40508  
{cheung,jkparu0}@engr.uky.edu

*Thin P. Nguyen*

EECS Department  
Oregon State University  
Corvallis, OR - 97330  
thinhq@eecs.oregonstate.edu

## ABSTRACT

Privacy protection of visual information is increasingly important as pervasive camera networks becomes more prevalent. The proposed scheme addresses the problem of preserving and controlling of the privacy visual data through two innovations. First, unlike the existing centralized control of privacy data, the proposed system allows individual users to make the final decision on every access to their privacy data. As such, it offers a much stronger form of privacy protection as the user no longer needs to trust, adhere or register his/her privacy preferences with a server. The second innovation is the development of a secure reversible data hiding scheme for embedding all the ownership information and privacy data into the obfuscated video bitstream. Not only has it resulted in an efficient design of protocols, the reversible data hiding allows perfect reconstruction of original data and supports arbitrary types of video obfuscation techniques. Impact of data hiding on bitrate and distortion is minimized through a rate-distortion optimization procedure and experimental results are provided to demonstrate its efficiency.

*Index Terms*— Privacy Protection, Reversible Data Hiding, Digital Right Management

## 1. INTRODUCTION

The right to privacy is one of the basic human rights implied by the Bill of Rights in the U.S. Constitution. In the last thirty years, advances in computing technologies have brought dramatic improvement in collecting, storing and sharing personal information among government agencies and private sectors. From the use of sophisticated pattern recognition in surveillance video to the theft of biometric and personal images or videos, more and more people have become increasingly weary about the privacy of their visual data. While new legislature and policy changes are needed to provide a comprehensive protection of personal privacy, technology is playing an increasingly important role to safeguard private information. While research on privacy protecting technologies have begun twenty years ago, most of the well-established schemes focus on textual or categorical data and are inadequate to protect visual information. As such, there have been a flurry of research activities in recent years to tackle the problem of privacy protection of visual information [1, 2].

These research focus on obfuscating sensitive information in the content but ignore the important issue of preserving and controlling the original data. Consider a video surveillance network in a hospital. While perturbing or obfuscating the surveillance video may conceal the identity of patients, the process also destroys the authenticity of the signal. Even with the consensus from the protected patients, law enforcement and arbitrators will no longer have access to

the original data for investigation. Thus, a privacy protection system must provide mechanism to enable users to selectively grant access to their private information. This is in fact the fundamental premise behind the Fair Information Practices [3, Ch.6]. In the near future, the use of cameras will become more prevalent. Dense pervasive camera networks are utilized not only for surveillance but also for other types of applications such as interactive virtual environment and immersive video-conferencing. Without jeopardizing the security of the organization, a flexible privacy data control system will become indispensable to handle complex privacy policy with large number of individuals to protect and different data requests to fulfil.

In this paper, we describe a system that provides secure and highly-flexible privacy data control for a pervasive camera network. We make two contributions compared with the state-of-the-art. First, unlike the existing centralized control of privacy data described in [4, 5], our system allows individual users to make the final decision on EVERY ACCESS to their privacy data. As such, our system offers a much stronger form of privacy protection as the user no longer needs to trust, adhere or register his/her privacy preferences with a server. Our second contribution is to develop a secure reversible data hiding scheme for embedding all the ownership information and privacy data into the obfuscated video bitstream. This scheme serves three different purposes – first, this allows a fully distributed implementation of the privacy data control with most of the complexity resided with the clients who need to access the privacy data. Second, the reversible data hiding allows a perfect reconstruction of the compressed data originated from the IP camera. When used with a secure camera with an authentication mechanism, such a reconstructed video can be authenticated and used as evidence in a court of law. Third, unlike privacy data preserving schemes in [6, 7] that can only be used with pixel scrambling, our data hiding approach puts no restriction on the obfuscation techniques. To minimize the impact on the distortion and bandwidth imposed by the data hiding process, we incorporate a rate-distortion optimization framework based on our earlier work for irreversible data hiding [8].

The rest of the paper is organized as follows. An overview of the system is provided in Section 2, followed by the description of the privacy data management in Section 3 and reversible data hiding scheme in Section 4. We present experimental results in Section 5 and conclude the paper in Section 6.

## 2. SYSTEM OVERVIEW

Before describing our proposed procedure for privacy data management and reversible data hiding, we first provide an overview of the entire system in which the proposed components are used. Implementation details of this system can be found in [9]. We call an indi-

vidual *user* if there is a need to protect his/her visual privacy. Video is stored as an individual segment of fixed duration with a unique ID that signifies the time and the camera from which it is captured. A *protected video* segment means that all the privacy information have been removed. A *client* refers to a party who is interested in viewing the privacy information of an user. Given these definitions, our system is designed to accomplish the following four goals:

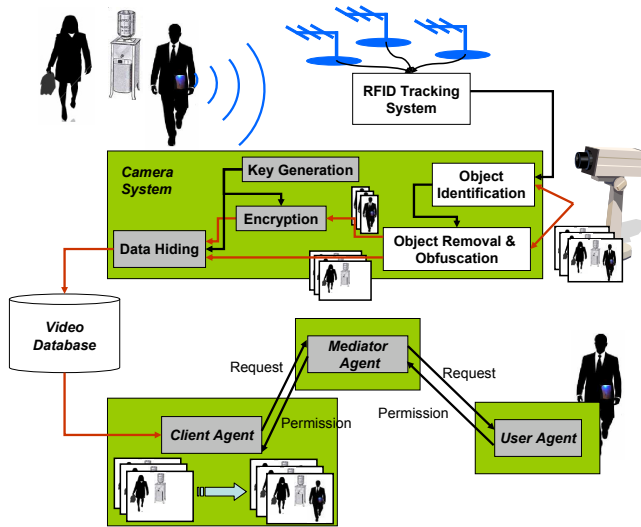
**Privacy** Without the proper authorization, a protected video and the associated data should provide no information on whether a particular user is in the scene.

**Security** Raw data should only be present at the sensors and at the computing units that possess the appropriate secret keys.

**Accessibility** A user can provide or prohibit a client's access to his/her imageries in a protected video segment captured at a specific time by a specific camera.

**Scalability** The architecture should be scalable to many cameras and should contain no single point of failure.

With these goals in mind, Figure 1 provides a high-level description of our system. Green boxes are secured processing units within which raw privacy data or decryption keys are used. All the processing units are connected through a open local area network, and as such, all privacy information must be encrypted before transmission. Red arrows show the flow of the compressed video and black arrows show the control information such as RFID data and key information. Each user in the environment is carrying an active RFID tag



**Fig. 1.** High-level description of our proposed privacy-protecting video surveillance system.

whose 2D location can be tracked real-time. After jointly calibrating the RFID system and the camera, the *Object Identification* module provides the bounding boxes of all users whose privacy needed to be protected [9]. While our current implementation of object identification module uses raw video, the rest of the system uses the compressed video in *motion-JPEG* format provided by the IP camera. Inter-frame compression is not used as keeping track of the embedded data inside a motion loop will make the modification irreversible. The *Object Removal and Obfuscation* module crops out the regions defined by the bounding boxes as privacy data, and replaces

them with pre-compressed background information for obfuscation. Other types of video obfuscation techniques can also be used here. The privacy data is encrypted at the *Encryption* module and embedded, with auxiliary data from the *Key Generation* module, into the obfuscated video bitstream at the *Data Hiding* module. The output protected bitstream is stored in the database and can be viewed by any standard-complaint player.

To achieve our goal of accessibility, the system must provide a mechanism for a client to obtain authorization from the user. This contradicts our privacy goal as the client has no way of knowing if a specific user is in the video. To address this problem, we propose a three-agent architecture by having a trusted *mediator agent* that relays request and permission between the *client agent* and the *user agent*. The protocols among these agents, the encryption and the key generation are all parts of a privacy data management system which is described in details in Section 3. The reversible data hiding scheme used in the data hiding module is explained in Section 4.

### 3. PRIVACY DATA MANAGEMENT

The goal of privacy data management is to allow individual users to control accessibility of their privacy data. This is reminiscent to a Data Right Management (DRM) system where the content owner can control the access of his/her content after proper payment is received. Our system is more streamlined than a typical DRM system as we have control over the entire data flow from production to consumption – for example, encrypted privacy information can be directly hidden in the protected video and no extra component is needed to manage privacy information. We use a combination of an asymmetric public-key cipher (1024-bit RSA) and a symmetric cipher (128-bit AES) to deliver a flexible and simple privacy data management system. RSA is used to provide flexible encryption of control and key information while AES is computationally efficient for encrypting video data. Each user  $u$  and client  $c$  publish their public keys  $PK_u$  and  $PK_c$  while keeping the secret keys  $SK_u$  and  $SK_c$  to themselves. As a client has no way of knowing the presence of a user in a particular video, there is a special *mediator m* to assist the client in requesting permission from the user. The mediator also has a pair of public and secret keys  $PK_m$  and  $SK_m$ .

Suppose there are  $N$  users  $u_i$  with  $i = 1, 2, \dots, N$  appeared in a video segment. We denote the protected video segment as  $V$  and the extracted video stream corresponding to user  $u_i$  as  $V_{u_i}$ . The Camera System prepares the following list of data to be embedded in  $V$ :

1.  $N$  AES-encrypted video streams  $AES(V_{u_i}; K_i)$  for  $i = 1, 2, \dots, N$ , each using a randomly generated 128-bit key  $K_i$ .
2. An encrypted table of content  $RSA(TOC; PK_m)$  using the mediator's public key  $PK_m$ . For each encrypted video stream  $V_{u_i}$ , the table of content  $TOC$  contains the following three data fields: a) the ID of user  $u_i$ ; b) the size of the encrypted bitstream; c) the RSA-encrypted AES key  $RSA(K_i; PK_{u_i})$  using the public key of the user and d) other types of meta-information about the user in the scene such as the trajectory of the user or the specific events involved the user. Such information helps the mediator to identify the video streams that match the queries from client. On the other hand, this field can be empty if the privacy policy of the user forbids the release of such information.

The process of retrieving privacy information is illustrated in Figure 2. When a client wants to retrieve the privacy data from

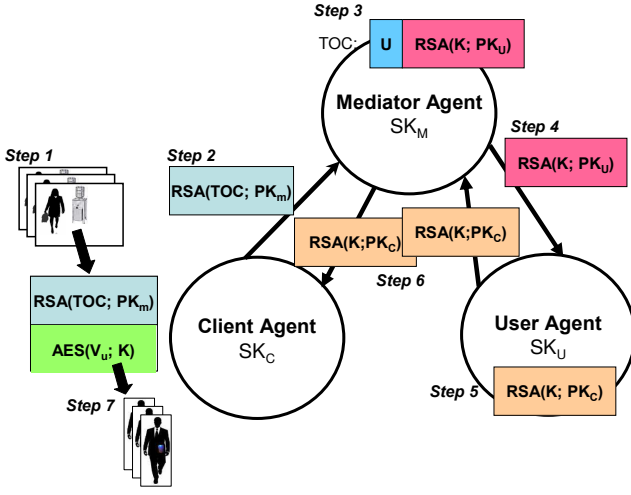


Fig. 2. Flow of privacy information.

a video segment, the corresponding client agent retrieves the hidden data from the video and extracts the encrypted table of content. The client agent then sends the encrypted table of content and the specific query of interest to the mediator agent. Since the table of content is encrypted with the mediator's public key  $PK_m$ , the mediator agent can decrypt it using the corresponding secret key  $SK_m$ . However, the mediator cannot authorize the direct access to the video as it does not have the decryption key for any of the embedded video streams. The mediator agent must forward the request to those users that match the client's query for proper authorization. The request data packet for user  $u_j$  contains the encrypted AES key  $RSA(K_j; PK_{u_j})$  and all the information about the requesting client  $c$ . If the user agent of  $u_j$  agrees with the request, it decrypts the AES key using its secret key  $SK_{u_j}$  and encrypts it using the client's public key  $PK_c$  before sending it back to the mediator. The mediator finally forwards all the encrypted keys back to the client which decrypts the corresponding video streams using the AES keys.

The above key distribution essentially implements a pseudo one-time pad for the encryption of each private video stream. As such, the decryption of one particular stream does not enable the client to decode any other video streams. The three-agent architecture allows the user to modify his/her private policy at will without first announcing it to everyone on the system. While the mediator agent is needed in every transaction, it contains no state information and thus can be replicated for load balancing. Furthermore, to prevent overloading the network, no video data is ever exchanged among agents. Finally, it is assumed that proper authentication is performed for each transaction to authenticate the identity of each party and the integrity of the data.

#### 4. REVERSIBLE DATA HIDING

Our data hiding process is performed at frame level so that the decoder can reconstruct the privacy information as soon as the compressed bitstream of the same frame has arrived. Privacy data is hidden only in the luminance Discrete Cosine Transform (DCT) blocks which typically occupy the largest portion of the bit stream. Our algorithm modifies the rate-distortion scheme introduced in our earlier work [8] to determine an optimal number of bits to be embedded

at each DCT block. As both the size of the bitstream and the decoded video quality using a standard-complaint player are adversely affected by the privacy data hidden in the bitstream, this optimization scheme is important to those clients who have no interest in retrieving the hidden privacy data. The rate distortion optimization framework depends on the underlying embedding process which we shall explain first.

The reversible embedding algorithm exploits the fact that DCT coefficients follow a Laplacian distribution concentrated around zero with empty bins towards either ends of the distribution [10]. Due to the high data concentration at the zero bin, we can embed high-volume of hidden data at the zero coefficients by shifting the bins right (or left) of zero to the right (or left).

At the encoder side, the embedding process is as follows: For the  $k$ -th quantized DCT block, the encoder first runs the rate-distortion optimization algorithm to determine that a specific number of bits, say  $M_k$ , is hidden in this DCT block. Let  $L = \lceil M_k/Z \rceil$  where  $Z$  is the number of zero coefficients in this DCT block. Following a reverse zigzag scan pattern starting from the highest frequency coefficient, we modify each DCT coefficients  $q(i, j, k)$  into  $\tilde{q}(i, j, k)$  using the following procedure until all the  $M_k$  bits of privacy data are embedded. Notice that we have  $i = 0, 1, \dots, 7$  and  $j = 0, 1, \dots, 7$  and  $k$  is the DCT block index.

1. If  $q(i, j, k)$  is zero, extract  $L$  bits from the privacy data buffer and set  $\tilde{q}(i, j, k) = q(i, j, k) + 2^{L-1} - V$  where  $V$  is the decimal value of these  $L$  privacy data bits.
2. If  $q(i, j, k)$  is negative, no embedding is done in this coefficient and  $\tilde{q}(i, j, k) = q(i, j, k) - 2^{L-1} - 1$ .
3. If  $q(i, j, k)$  is positive, no embedding is done in this coefficient but  $\tilde{q}(i, j, k) = q(i, j, k) + 2^{L-1}$ .

Note that the maximum distortion at each coefficient increases from one QP (quantization parameter) to  $(2^L + 1)QP$  due to this embedding.

On the decoder side, it needs to extract the hidden bits and retrieve the original quantized coefficient  $q(i, j, k)$  from  $\tilde{q}(i, j, k)$ . The decoder also knows the number of hidden bits  $M_k$  by running the same rate distortion algorithm. Starting at the highest frequency coefficient in a reverse zigzag pattern, the privacy data and the original DCT coefficient can be obtained as follows:

1. If  $-2^{L-1} < \tilde{q}(i, j, k) \leq 2^{L-1}$ ,  $L$  hidden bits can be obtained as the binary equivalent of the decimal number  $2^{L-1} - \tilde{q}(i, j, k)$  and  $q(i, j, k) = 0$ .
2. If  $\tilde{q}(i, j, k) \leq -2^{L-1}$ , no hidden bit in this coefficient and  $q(i, j, k) = \tilde{q}(i, j, k) + 2^{L-1} - 1$ .
3. If  $\tilde{q}(i, j, k) > 2^{L-1}$ , no bit is hidden in this coefficient and  $q(i, j, k) = \tilde{q}(i, j, k) - 2^{L-1}$ .

We now come back to our rate-distortion optimization algorithm. Let  $D$  and  $R$  be the distortion and bit-rate increase caused by hiding  $N$  bits of privacy data in the bitstream. By using a user-specified control parameter  $\delta$ , we combine the rate and distortion into a single cost function as follows:

$$C = (1 - \delta) \cdot N_F \cdot D + \delta \cdot R \quad (1)$$

$N_F$  is used to normalized the dynamic range of  $D$  and  $R$ .  $\delta$  is selected based on the particular application which may favor the least amount of distortion by setting  $\delta$  close to zero, or the least amount of bit rate increase by setting  $\delta$  close to one. To practically solve for the above optimization, we assume the cost function in (1) can be written as the sum from individual blocks, and both the distortion

and the rate increase at each block  $k$  are functions of the number of bits embedded  $M_k$ :

$$C_k(M_k) = (1 - \delta) \cdot N_F \cdot D_k(M_k) + \delta \cdot R_k(M_k) \quad (2)$$

The overall optimization then becomes

$$\min \sum_k C_k(M_k) \quad \text{subjected to} \quad \sum_k M_k \geq N \quad (3)$$

As high frequency coefficients have lesser impact on perceptual quality, we heuristically set the embedding to start from the zero coefficient of the highest frequency and follow a reverse zigzag scan pattern in measuring  $D_k(M_k)$  and  $R_k(M_k)$ . The models are built using the previous frame so the same process can be carried out in both the encoder and the decoder. The distortion  $D_k(M_k)$  is based on a perceptual distortion metric defined using DCT coefficients as in [8] and the rate increase is directly calculated by embedding a '0' bit sequence onto the DCT block using the run-length and entropy coding as in the actual compression process. Due to the difference in the embedding process, the computation of  $D_k(M_k)$  and  $R_k(M_k)$  are different from those in [8] but the overall optimization can be solved using a similar Lagrangian technique.

## 5. EXPERIMENTS

We tested our reversible data hiding algorithm on the "hall monitor" test sequence which has 299 frames in CIF format ( $352 \times 288$ ). The entire video is compressed in INTRA mode using a regular H.263 encoder. The macroblocks of the person on the right are extracted as privacy data and replaced by an adaptive estimate of the background. Key information and privacy data are then embedded using our reversible data hiding scheme with  $\delta = 0.5$ . The first table shows the bit-rates of the obfuscated stream ( $R_o$ ), the privacy data ( $R_p$ ) and the obfuscated stream with embedded data ( $R_e$ ) at different QP's. Depending on the QP, we see that the data embedding process incurs an expansion of bitrate from 26% to 58%, which are inline with the measurements we obtain for irreversible data hiding [8]. Figure 3 shows one sample frame of the sequence at QP=10 decoded by a reversible and regular decoders respectively. The second table shows

Bit rates (kbps) of reversible embedding				
QP	$R_o$	$R_p$	$R_e$	% increase
20	1560	710	3607	58.9
15	1885	744	3845	46.2
10	2493	807	4656	41.1
5	4018	960	6281	26.2

the results of the same sequence at QP=10 under different values of the control parameter  $\delta$  in the optimization process. The results show that we have the option of trading-off bitrate with perceptual distortion by using different values of  $\delta$ .

Optimization results embedding at QP=10 (807 kbps)		
$\delta$	Rate	Perceptual Distortion
Original	2493 kbps	0
0	4757 kbps	127
0.5	4656 kbps	135
1	3856 kbps	255



**Fig. 3.** 125th frame of Hall Monitor Sequence at QP = 10 decoded by reversible and regular decoders.

## 6. CONCLUSIONS

In this paper, we have proposed a flexible privacy data management system that uses a three-agent architecture for exchanging key information and a reversible data hiding scheme for combining key and privacy data with obfuscated bitstream. Larger scale experiments are still needed to test the robustness of the data management scheme. We are also investigating alternative coding structure and optimization procedure to mitigate the increase of bitrate caused by the data hiding process.

## 7. REFERENCES

- [1] A. Senior, S. Pankanti, A. Hampapur, Y.-L. Tian L. Brown, and A. Ekin, "Blinkering surveillance: Enabling video privacy through computer vision," *Security and Privacy*, vol. 3, pp. 50–57, 2005.
- [2] J. Schiff, M. Meingast, D. Mulligan, S. Sastry, and K. Goldberg, "Respectful cameras: Detecting visual markers in real-time to address privacy concerns," in *International Conference on Intelligent Robots and Systems (IROS)*, 2007.
- [3] D. J. Solove, *The Digital Person: Technology and Privacy in the Information Age*, New York University Press, 2004.
- [4] G. V. Lioudakis et al., "A middleware architecture for privacy protection," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 51, no. 16, pp. 4679–4696, November 2007.
- [5] D.-A. Fidaleo, H.-A. Nguyen, and M. Trivedi, "The networked sensor tapestry (nest): a privacy enhanced software architecture for interactive analysis of data in video-sensor networks," in *VSSN '04: Proceedings of the ACM 2nd international workshop on Video surveillance & sensor networks*, New York, NY, USA, 2004, pp. 46–53, ACM Press.
- [6] Frdric Dufaux and Touradj Ebrahimi, "Scrambling for video surveillance with privacy," *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, p. 160, 2006.
- [7] T. E. Boulton, "Pico: Privacy through invertible cryptographic obscuration," in *Proc. Computer Vision for Interactive and Intelligent Environments: The Dr. Bradley D. Carter Workshop Series*. University of Kentucky, 2005.
- [8] J. K. Paruchuri and S.-C. Cheung, "Joint optimization of data hiding and video compression," in *IEEE International Symposium on Circuits and Systems (ISCAS 2008)*, 2008.
- [9] M. V Venkatesh, S.-C. Cheung, J. K. Paruchuri, J. Zhao, and T. Nguyen, *Protecting and Managing Privacy Information In Video Surveillance Systems.*, To appear in "Protecting Privacy in Video Surveillance", Springer, 2008.
- [10] C. C. Chang, W. L. Tai, and M. H. Lin, "A reversible data hiding scheme with modified side match vector quantization," in *Proceedings of the International Conference on Advanced Information Networking and Applications*, 2005, vol. 1, pp. 947–952.