EVALUATION OF JPEG2000 HASHING FOR EFFICIENT AUTHENTICATION

Dominik Engel, Thomas Stütz, and Andreas Uhl

Department of Computer Sciences
University of Salzburg, Austria
{dengel,tstuetz,uhl}@cosy.sbg.ac.at

ABSTRACT

In this paper we investigate the applicability of different parts of the JPEG2000 codestream for authentication. Apart from the packet body different classes of information contained in the packet header are investigated. We report on experimental results obtained with a large test set of natural images to assess how discriminative and how sensitive each class of information is. Specific attacks against authentication schemes, that use selective hashing of either packet bodies (as proposed in literature) or packet headers, are presented.

1. INTRODUCTION

The need for authentication of JPEG2000 content has been addressed in various forms, and a number of suggestions have followed the first proposal for an authentication scheme by [1]. With the finalization of part 8 of the JPEG2000 standard, JPSEC [2], there is even a standardized way to authenticate JPEG2000 content. JPSEC provides tools and solutions for securing JPEG2000-coded visual content. It extends the JPEG2000 codestream syntax to implement security services, e.g., encryption and authentication. Furthermore, complementing the normative part, informative examples for the application of the standard are given. For scalable authentication, JPSEC allows to hash and sign different parts of the bitstream at different granularity levels.

In this paper we evaluate the utility of different parts of JPEG2000 packets for scalable hashing schemes which can be incorporated into the JPSEC framework. We compare the utility of whole packets and the packet payload contained in the packet body that have traditionally been used to the utility of different parts of the JPEG2000 packet headers. The rationale for this is the observation that the JPEG2000 packet header content is distinctive for the *visual* content of the packet – each packet header can be conceived as a hash for the packet body. It has been pointed out that the packet header is specific enough for content search and can be used as a distinctive fingerprint [3]; we will investigate here if it is also specific and secure enough for the purpose of authentication.

As the packet headers make up less than 2% of the bitstream (for usual settings, see Section 3) signing only these data can severely reduce the computational requirements and can therefore make a decisive difference for authentication in an environment of low-end processors, such as mobile devices with low processing power. Of course, the question will have to be addressed, if using the header alone opens the door for new attacks.

This paper is organized as follows: in Section 2 we introduce prior and related work. Section 3 gives a brief overview of authentication in the context of JPSEC. As the packet headers have so far not been investigated as a processing domain in their own rights, in Section 4 we discuss the classes of information contained in the packet header and their possible utility in an authentication scheme. In Section 5 we discuss attacks against selective hashing for both packet header and packet body based schemes. Section 6 summarizes the results and concludes.

2. RELATED WORK

There are two major classes of image hashing schemes, based on the intended application: authentication and CBIR hashes (content-based image retrieval). Authentication hashes are used to authenticate visual data, and register if visual data has been tampered with. For this kind of hashes some robustness is desired against tolerated image modifications, but far less than in the case of retrieval hashes. Furthermore, there is the requirement to reliably detect malicious alterations of the image content.

In the context of authentication hashes, many suggestions for robust hashes have been put forward in recent years, e.g., [4, 5]. Some proposals have been made for JPEG2000 hashing and authentication. [1] propose to apply a standard SHA-1 onto all packet data and to append the resulting hash value after the final termination marker to the JPEG2000 bitstream. This scheme constructs a 160-bit SHA-1 hash for each codeblock which is then encrypted with RSA. This scheme protects only data that is contained in the packet bodies. Therefore this scheme is susceptible to the attacks against packet body hashing presented in this work.

[6] propose an elegant solution to scalable authentication of JPEG2000 using Merkle hash trees. The hash is created by hashing whole packets. The different schemes on different

Codestream entity	Length in bytes
Avg. main header	130.00
Avg. packet header length	6.52
Avg. packet body length	409.03
Avg. packet length	434.66
Avg. number of packets	153.41

Table 1. Average length of the different codestream entities

granularities we discuss here, e.g., based on the JPEG2000 packet headers, can be used for the creation of the Merkle hash trees by substituting the hashing method.

[7] propose a robust hashing scheme based on JPEG2000 that employs parts of the packet body: starting at the beginning of the codestream in layer progression packet body data up to a specified length is used as a hash. Therefore this scheme is susceptible to the attacks against packet body hashing presented in this work.

3. JPEG2000 BITSTREAM AUTHENTICATION WITH JPSEC

In many applications, authentication is the most important security service. Even when confidentiality is the targeted security service, it should be augmented by authentication to prevent attacks [2, p.29]. Therefore an authentication template exists in JPSEC (a normative tool). Three authentication methods are standardized: Hash-based MACs, Cipher-based MACs and Digital Signatures. These authentication methods can be applied on different granularity levels. JPSEC allows to specify very freely how to process (authenticate in our application) the JPEG2000 data. JPSEC specifies several syntactic/semantic elements that ensure the unique processing of JPEG2000 data. The "Zone of Influence" (ZoI) specifies which JPEG2000 parts are subject to further processing. These fractions of the JPEG2000 codestream can be defined by image and non-image related parameter classes [2, p.16]. It is, e.g., possible to specify that only resolution 0 and 1 are authenticated. The "processing domain" specifies in which domain the tool is applied (e.g., pixel domain, wavelet domain, codestream domain). Furthermore the processing domain allows to specify if the protection method is applied to the packet header and the packet body or just the packet body. Note that the packet headers on their own are not a standard processing domain in the normative part of JPSEC. To incorporate the packet header based hash into JPSEC the definition of a user defined tool or ZoI is necessary. The granularity level specifies in which granularity the tool is applied (e.g. on tiles, components, resolutions, layers, precincts, packets).

In order to estimate the cost for authentication at the different granularities, in Table 1 we give the average length of packet header, packet body and whole packet. The numbers were computed for each of the 1000 images in the test set with the coding settings described in Section 4. Note that the selective hashes we discuss here are all on the granularity of packet level (one hash is produced for each packet). As every hash or HMAC value can be expected to have a size of at least 16 byte, authentication at a fine granularity can have a significant impact on compression performance. Processing on a packet basis increases the file size about 3.37%, while processing on resolutions results in a file size increase of 0.14%. Thus in applications it may be necessary to restrict the granularity to resolution level, i.e., construct only one hash for all the contributions to a certain resolution level. Table 1 also shows that the average packet headers make up only 1.49% of the codestream. If the packet header has a high level of discriminative power, then a selective hashing scheme that only hashes or even directly signs the packet headers can save a lot of computational effort. Note that only hashing 2% of the packet body data is not a feasible approach as such a hash would only pertain to the first codeblocks in the packet and not register changes in the rest.

4. JPEG2000 PACKET HEADERS AS A DISCRIMINATIVE FEATURE

The packet headers contain information on different properties of the packet. We will refer to each different kind of header information as a "class" in the following. The classes that can be used for hashing are: the packet header length (hLEN), the packet body length (bLEN), the length of the contribution of each codeblock to the packet (CCP), the number of leading zero-bitplanes (LZB), the inclusion information of each codeblock (INC), and the number of coding passes that are contained in the packet for each codeblock (NTP). We add three more classes: hSHA, bSHA, and pSHA, which are 160-bit SHA-1 hash values for all of the packet header data, all of the packet body data and for all of the whole packet data, respectively. We investigate all of these classes regarding their discriminative power, i.e., their utility in telling different images apart.

The 1000 images of our test set have a size of 512×512 pixels and are 8bpp grayscale. We compute the header hash for each class of header information for each packet in each image. To determine how many packets we need to reliably tell the images apart, we compare the hashes at all possible lengths of each possible pair among the 1000 images and record exact matches. We use a wavelet decomposition level of 5, 32 quality layers in resolution progression order and a codeblock size of 64. These settings produce a maximum of 192 packets for each image.

Our evaluations reveal that the packet header information of the first 10 packets has the same discriminative power as information of all the packets (192) . Furthermore it turns out that CCP, NTP, INC and LZB have different discriminative properties. For the LZB more than 120 packets are needed to reliably authenticate an image in the test set. The other

(a) Tamper Mode 1 (b) Tamper Mode 2

Fig. 1. Example for tampered images

three header information classes exhibit higher discriminative power, with the CCP being the most discriminative feature.

In terms of discriminative power we can summarize that a hash over the packet header works as well as a hash of the whole packet and that even for a small number of packets. Single classes of the header information could also be used, but reduce the discriminative power in different degrees, compared to the combination of all features.

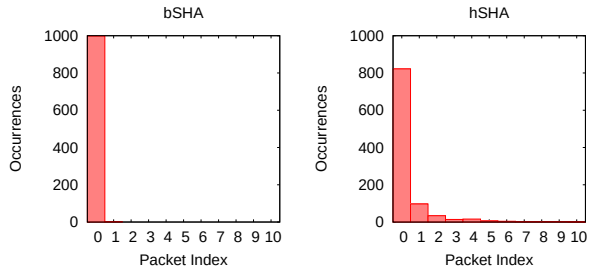
4.1. Sensitivity

Any authentication scheme needs to be sensitive towards malicious alterations. We have already shown that after relatively few packets in terms of distinguishing different images the selective hashing schemes have the same discriminative power as hashing schemes based on the whole packet. In the following we investigate if they also exhibit the same sensitivity.

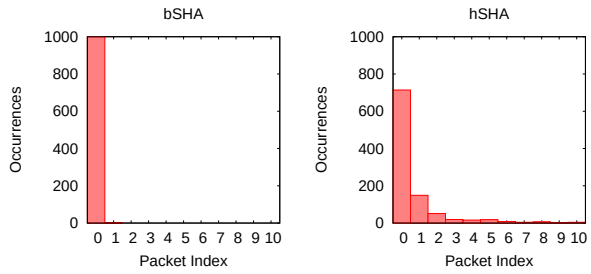
We quantitatively investigate sensitivity with the set of 1000 images by tampering with each of the images. We use two types of tampering, in both of which the tampered images are marked at random positions. In mode 1, the text "tampered" is inserted, in mode 2 a white "x" followed by a black "o" is inserted. Examples are shown in Figure 1. We use resolution progression order. The sensitivity results are shown in Figure 2. The plots show the packet index for which the modification is first registered on the abscissa. The ordinate shows the number of images. It can be seen that the packet body hash reliably detects both of the modifications after 2 packets. The header hash needs more packets, especially for the shorter modification, but also succeeds in detecting the tampering in both cases in Resolution 0 (which corresponds to a 16×16 pixel version of the original image).

5. ATTACKS AGAINST SELECTIVE HASHING

The results of the previous sections indicate that both, the packet header and the packet body data are suitable for image authentication. Packet header based schemes can lower computational demands, as those parts are only small fraction of the overall data. In this section we investigate what tradeoff in terms of security has to be expected. Hashing only the



(a) Tamper Mode 1



(b) Tamper Mode 2

Fig. 2. Histograms for sensitivity comparing header and body hash

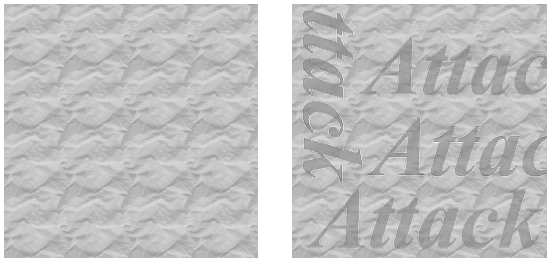
JPEG2000 packet bodies can be implemented in JPSEC (but definitely should not!).

5.1. Selective Packet Header Hashing

An important question is if, given an image, another image – visually distinct from the original – with the exact same hash can be constructed. It is absolutely possible to rewrite a packet body that fits a given packet header, as numerous proposals for format-compliant encryption of packet bodies show (that leave the packet header in plaintext)[3]. Packet body data can be selectively encrypted, thereby hiding the image content. The encryption can be applied with fine granularity, e.g., on single coding passes of a codeblock. This method enables an attacker to hide arbitrary portions of the codestream without affecting the JPEG2000 packet header. A signature on the packet header hash remains valid for the codestream, even if the packet bodies are later reverted to their original state. The visual examples for this attack look similar to the results for the selective packet body hashing attack (as shown in Figure 3).

5.2. Selective Packet Body Hashing

There are practical attacks against schemes that only hash (and sign) the packet body. As an example an image was created that in its wavelet representation contains an inconspicuous image in the lower 4 resolutions. The coefficients of the 5^{th} resolution were replaced with the coefficients of an image containing only the word "Attack". In the JPEG2000



(a) Reconstruction of JPEG2000 codestream with transformed headers

(b) Reconstruction of JPEG2000 codestream with non-transformed headers

Fig. 3. Illustration of attack against selective packet body hashing

compressed domain of this image, we applied a transformation to the number of leading zero bitplanes, more precisely we added 220 modulo 256 to each of number of LZB. Thus these coefficients are basically treated as zero and their image information (the word "Attack") is not visible (as illustrated in Figure 3(a)). Nevertheless the signature for such an image is also valid for the reconstruction of a JPEG2000 codestream with non-transformed headers, that is shown in Figure 3(b). The word "Attack" is clearly visible and the hash is the same for both images and thus the signature (signed hash) is valid for both images.

5.3. Applicability of selective hashing

The above analysis clearly shows that the target application of a hash restricted to parts of a packet can never be high security applications of authentication. Neither packet header nor packet body should be used for that end. This point is even more important for authentication schemes that are based on the packet body, where packet header rewriting has to be seen as a novel threat.

Hashes restricted to the packet header present an efficient way for integrity checking, but are not secure against specifically tailored attacks.

6. CONCLUSION

Our experimental results indicate that packet header data offer a high level of discriminative power and a high sensitivity to tampering attacks. This observation brings the possibility for efficient authentication in application environments with clients of low processing powers, however, at the cost of security, which has to be traded in for the increased efficiency.

Furthermore our work reveals that packet body based authentication schemes, as proposed in literature and as can but should not be implemented in the JPSEC framework, are vulnerable to similar attacks. Therefore schemes that restrict hashing to the packet body are not secure and shall not be used for authentication purposes. A practical attack against these schemes is presented. In summary, the most secure option is hashing of the entire JPEG2000 packet. With respect to compression performance even a coarser granularity is beneficial. If security has to be traded of for efficiency, packet header hashing can be an option.

Future work should focus on the construction of JPEG2000 hashing schemes that are robust against common, but tolerable image modifications, such as baseline JPEG compression. The discussed approaches do not have this property.

7. REFERENCES

[1] Raphaël Grosbois, Pierre Gerbelot, and Touradj Ebrahimi, "Authentication and access control in the JPEG2000 compressed domain," in *Applications of Digital Image Processing XXIV*, A.G. Tescher, Ed., San Diego, CA, USA, July 2001, vol. 4472 of *Proceedings of SPIE*, pp. 95–104.

[2] ISO/IEC 15444-8, "Information technology – JPEG2000 image coding system, Part 8: Secure JPEG2000," Apr. 2007.

[3] Dominik Engel, Thomas Stütz, and Andreas Uhl, "Format-compliant JPEG2000 encryption in JPSEC: Security, applicability and the impact of compression parameters," *EURASIP Journal on Information Security*, vol. 2007, no. Article ID 94565, pp. doi:10.1155/2007/94565, 20 pages, 2007.

[4] Jiri Fridrich and Miroslav Goljan, "Robust hash functions for digital watermarking," in *Proceedings of the IEEE International Conference on Information Technology: Coding and Computing*, Las Vegas, NV, USA, Mar. 2000.

[5] V. Monga, A. Banerjee, and B. L. Evens, "A clustering based approach to perceptual image hashing," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 1, pp. 68–79, 2006.

[6] Cheng Peng, Robert Deng, Yongdong Wu, and Weizhong Shao, "A flexible and scalable authentication scheme for JPEG2000 codestreams," in *Proceedings of ACM Multimedia 2003*, San Francisco, CA, USA, Nov. 2003, pp. 433–441.

[7] R. Norcen and A. Uhl, "Robust visual hashing using JPEG2000," in *Eighth IFIP TC6/TC11 Conference on Communications and Multimedia Security (CMS'04)*, D. Chadwick and B. Preneel, Eds., Lake Windermere, GB, Sept. 2004, pp. 223–236, Springer-Verlag.