Active Quotation Authentication in Microsoft Word Documents using Block Signatures

Tsung-Yuan Liu[1] and Wen-Hsiang Tsai[1, 2]

[1]Department of Computer and Information Science
National Chiao Tung University
Hsinchu, Taiwan

[2]Dept. of Computer Science and Information Engineering
Taichung Healthcare and Management University
Taichung, Taiwan

Abstract **Traditional message authentication does not verify the integrity and the claimed source of quotations in an anonymous message. A new active quotation authentication framework is proposed that allows a reader to authenticate the content and origin of the quotations in a citing document. A block signature method is proposed that generates context-sensitive block signatures for an authoritative message, and allows quotation signatures to be generated on-the-fly for arbitrary sub passages by an anonymous author. The block signature method requires a small initial computational effort and storage requirement for block signature generation; has a small quotation signature overhead; and is efficient for quotation authentication.**

Index Terms**— Message authentication, active quotation authentication, Internet hoax.**

I. Introduction

The proliferation of the Internet eased the publication of information to the mass, with a growing number of authors providing information that may be of interest to a particular user group in the form of web pages, e-mails, or electronic documents. The wealth of contents made available is mostly beneficial to the general public, but there is also abundance of annoying or even harmful information.

Internet hoaxes such as sympathy letters and virus warnings, for example, are often forged, but by using the *credibility by association technique* [1], people tend to take a degree of belief in the message. One mobile phone virus hoax, for example, claims that if a user answers a call without caller identity, the user's phone can be affected by the virus and will not be usable any more. The hoax lures the reader by claiming that the information has been confirmed by credible mobile phone companies (Motorola and Nokia), and that the news was reported on CNN. Due to the credibility of the claimed sources, most people will not perform any further verification and will pass the message on to others, regardless of the credibility of the sender. The multiplicative effect of the blind message passing is alarming, putting unnecessary burdens on the network and servers, and costs the reader valuable time to process them.

This work was supported partially by the NSC Advanced Technologies and Applications for Next Generation Information Networks (II) – Sub-project 5: Network Security, Project No. NSC93-2752-E-009-006-PAE.

Message authentication research concentrated on verifying the integrity and source of a message as a whole. HMAC [2, 3] and CBC-MAC [4] uses a shared secret key between two communicating parties to validate the transmitted content over insecure channels. Digital signatures [5, 6] use asymmetric cryptography to sign a message digest of the transmitted content. Although theses researches, with some prior arrangements on the cryptographic keys, can validate the authenticity of the transmitted messages, it falls short for the problem described previously. Specifically, the authentication mechanisms can only verify that a particular message is indeed sent by a trusting party, but cannot reveal whether the contents within are from authoritative sources.

In this paper we propose a new technique of message authentication, called *active quotation authentication*, where quotations from authoritative sources are quoted along with the necessary authentication information, with the result that the final reader can verify the validity of such quotations within an anonymous message. The idea of the proposed active quotation authentication technique is described in Section 2, and in Section 3, a new block signature method is proposed, which has efficient signature generation and authentication performance. Section 4 describes the experiments, and Section 5 concludes the paper with some suggestions for future works.

II. Active Quotation Authentication

Quotation authentication, as studied in this work, aims to provide verification of the integrity and identify the source of quotations in a message. In this work, we consider quotations as consecutive characters in a message. Let M be a *public message* which includes several quotations denoted by $q_1, q_2, \ldots, q_N$, where each q_i is cited from a certain *credible source message* R^i authored by a certain person A^j, as shown in Fig. 1. For simplicity, we will mention M simply as a *message* and R^i as a *source* in the sequel. The main goal of quotation authentication is to verify the integrity of each quotation q_i in a message M, and to identify the author A^j of the source R^i of q_i.

For each source R^i, we assume that the author A^j signs it with his/her private key K^j using one of the signing methods described in the next section. The author A^j has a corresponding public key PK^j, which is to be used by the final reader to verify

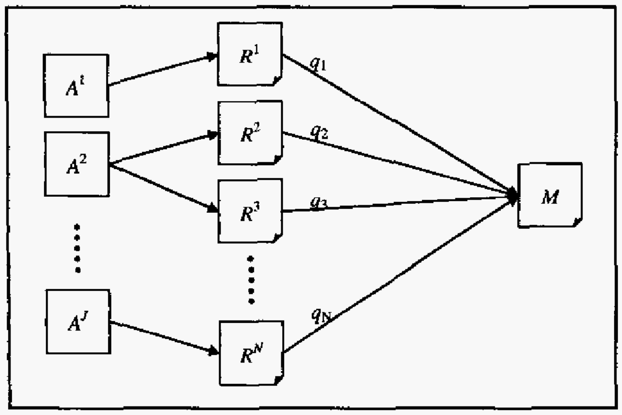


Figure 1. Including quotations from sources $R^1 \ldots R^N$ by authors $A^1 \ldots A^J$ into public message M.

that the source R^i has been signed by A^j using K^j. We assume that there is a globally trusted party (GTP), and that each author A^j has previously registered his/her public key PK^j with the GTP. An appropriate digital certificate (a group of digital data including the author's name, public key, and so on) is issued by the GTP to prove the registration process. Digital certificates such as those defined in ITU-T X.509 [7] can be used for this purpose.

We have chosen the Microsoft Word document as the message interchange media, due to its popularity and powerful features. Microsoft Word allows plug-ins to be installed into the Microsoft Word environment, expanding the word processor's ability in user specific ways. An authentication plug-in that uses the proposed approach can let a credible-source author sign his/her source document with a click of a button.

In more details, when a credible-source author signs a Word document, one or more *source signatures* are generated, as described in the next section. The signatures and the author's digital certificate are embedded into the document, along with a VBA Macro [8], which allows a public-message author to select as a *quotation* any passage in the signed source and use a key stroke (such as Alt-C) to actively compute an appropriate signature for the quotation, called *quotation signature*.

The quotation along with the signature is copied and pasted, using the VBA Macro, into a public-message author's document as an *authenticated quotation*. Finally, the authentication plug-in provides the verification functionality, which allows the final reader to easily verify the integrity and identify the source of the authenticated quotations in a message by clicking a verification button.

III. Source and Quotation Signatures

For each source R^i, the author A^j creates one or more source signatures using his/her private key K^j with one of the methods described below. When a public-message author selects a passage q in R, he/she dynamically generates a quotation signature p for that quotation. The quotation signature includes one of the source signatures, the author's digital certificate, and some additional auxiliary information, so that a final reader can verify the integrity of the quotation and the fact that it is authored by A^j. We assume that there is a one-way collision-resistant hash function $H(\cdot)$, which reduces a message of arbitrary length to a compact message digest. We also assume that there is a message signing function $S_K(\cdot)$, and a corresponding signature verification function $V_{PK}(\cdot)$, such that $V_{PK}(M', S_K(M))$ is *false* for all $M \neq M'$, and *true* for all $M=M'$ for any matching private key-public key pair. We present and discuss two ways of generating the source and quotation signatures for the purpose of active quotation authentication.

A. Whole Message Approach

In this method, only one source signature is generated by the author by signing the entire source message R with his/her secret key K to arrive at the source signature $P = S_K(R)$. When a public author wishes to quote some texts q that is inside the source R, he/she constructs the quotation signature p simply by including R, P, and the author's digital certificate. For the final reader to verify the authenticity of the quotation q in question, he/she first extracts R, P, and the author's digital certificate from p, and verify that the message is intact by the verification function $V_{PK}(R, P)$. Finally, a check is performed to verify that q is a substring of R.

A clear disadvantage of the above method is in including the entire source R for a quotation q, as the length of a quotation is typically much less than that of the source message. By including many quotations in a message M, the quotation signatures required using this method will increase the size of the resulting message considerably.

B. Block Signature Approach

We propose a new block signature method, where the source message is partitioned into blocks of characters, and a context-sensitive source signature is generated for each block of texts. The quotation signature then contains one of the source signatures and some other supplementary information. The result of this approach is that the size of the quotation signature for any quotation can be made small, and independent of the length of the source message or the quotation. We describe in detail the source signature generation, the quotation signature generation, and quotation authentication in the sequel. The method for generating source signatures is illustrated in Fig. 2 and described below.

Algorithm 1: Source signature generation for active quotation authentication.

Input: A source message R containing X characters, $c_1, c_2, \ldots, c_X$, and block size N that is an input variable.

Output: The number of blocks Y, the context-sensitive *block-chain hash codes* $H_1, H_2, \ldots, H_Y$, and source signatures $S_1, S_2, \ldots, S_Y$.

Step 1. Partition the source into $Y = \lceil X/N \rceil$ blocks of characters, with each block containing N characters except the last block. Denote the blocks of characters as $B_1=\{c_1, c_2, \ldots, c_N\}$, $B_2=\{c_{N+1}, c_{N+2}, \ldots, c_{2N}\}$, ..., $B_Y=\{c_{(Y-1)N+1}, c_{(Y-1)N+2}, \ldots, c_X\}$.

Step 2. Compute the hash value of block one as the first

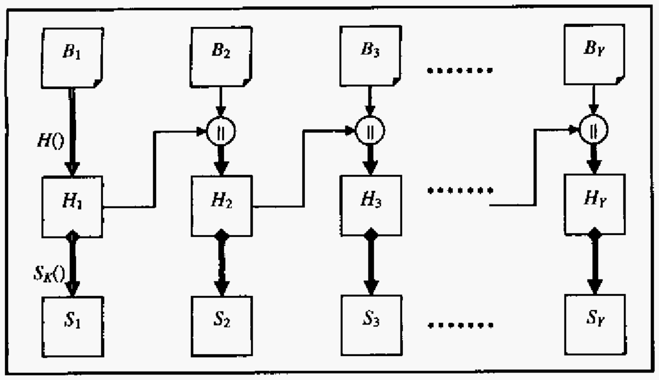


Figure 2. Generating source signatures in the proposed method.

block-chain hash code, that is, compute $H_1=H(B_1)$. Compute the first source signature $S_1=S_K(H_1)$ as the signature of H_1.

Step 3. For each integer i from 2 to Y,

a. Concatenate the previous block-chain hash code with the current block of characters, and then calculate the hash value of the concatenation as the next block-chain hash code, that is, $H_i=H(H_{i-1} \| B_i)$ where "||" means concatenation.

b. Compute the signature of the block-chain hash code H_i as the source signature, that is, $S_i=S_K(H_i)$.

The above source signature generation algorithm generates $Y=\lceil X/N \rceil$ signatures for a source R of X characters. Each source signature requires: 1) the previous block-chain hash code (except the first) to be concatenated with the current block of characters; 2) the hash value of the concatenation to be computed; and 3) the signature for the hash value to be generated. The result of the chaining process is a context-sensitive signature – the signature is dependant on every block of characters before it. Source signature S_k can thus be used to verify the integrity of the concatenation $B_1 \| B_2 \| \ldots \| B_k$ for any $1 \le k \le Y$. We note that the computational effort to generate a source signature is essentially constant, and thus the total computational effort to generate all source signatures for a source message is linear to the source length.

The quotation signature is generated dynamically when an author copies a quotation from the signed source. We first consider the simple case where a quotation contains *full* blocks of characters, that is, a quotation is the character sequence formed by concatenating the character blocks $B_j \| B_{j+1} \| \ldots \| B_k$ for some $1 \le j \le k \le Y$. In this case, the quotation signature includes the source signature S_k, and a block-chain hash code H_{j-1} if $j > 1$. The above quotation signature is sufficient and necessary because the block-chain hash codes H_j, H_{j+1}, …, H_k can be reconstructed from the quotation texts and the starting block-chain hash code H_{j-1}, that is, $H_j=H(H_{j-1} \| B_j)$, $H_{j+1}=H(H_j \| B_{j+1})$, and so on. By verifying that the source signature S_k is the correct signature for the reconstructed hash code H_k, we confirm that the entire quotation is intact. Due to the collision-resistant requirement of the hash function, it will be hard for an adversary to create a fraudulent quotation and some starting block-chain hash code that will result in a hash code H' matching H_k.

In general, a quotation may not contain full blocks of characters, that is, a quotation may be any character sequence c_a, c_{a+1}, …, c_b in the source, where a and b are any integer such that $1 \le a \le b \le X$. The algorithm to generate the quotation signature for the general case is presented below.

Algorithm 2: Quotation signature generation for the purpose of quotation authentication.

Input: Signed source R containing X characters, $c_1, c_2, \ldots, c_X$, and partitioned into $Y=\lceil X/N \rceil$ blocks $B_1, B_2, \ldots, B_Y$, where N is the block size, the context-sensitive block-chain hash codes $H_1, H_2, \ldots, H_Y$, and source signatures $S_1, S_2, \ldots, S_Y$. The quotation $q = c_a, c_{a+1}, \ldots, c_b$, where $1 \le a \le b \le X$, for which the quotation signature is to be generated.

Output: Starting and ending block indices j and k. The quotation signature p which contains the block signature S_k, the starting block-chain hash code H_{j-1} (if $j > 1$), and *filling* blocks B'_j and B'_k where necessary (these are used to pad the quotation to full blocks of characters).

Step 1. Set j and k such that the concatenated character blocks $B_j \| B_{j+1} \| \ldots \| B_k$ just contain the quotation texts q. Specifically, set $j=\lfloor (a-1)/N \rfloor+1$, and set $k=\lfloor (b-1)/N \rfloor+1$.

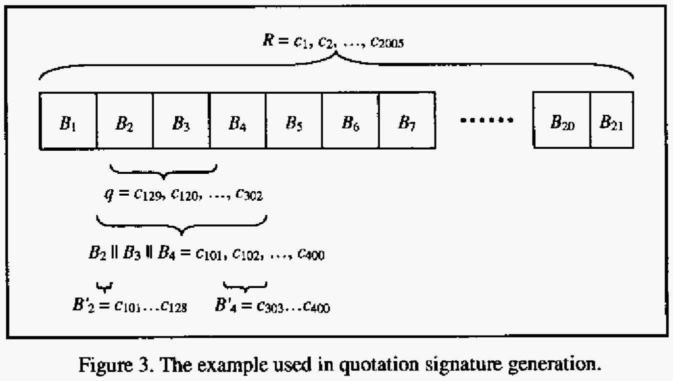
Step 2. Output starting block-chain hash code H_{j-1} if $j > 1$.

Step 3. Output the prefix block $B'_j = c_{(j-1)N+1}, c_{(j-1)N+2}, \ldots, c_{a-1}$ if a is not the start of block B_j, that is, if $a \ne (j-1)N+1$.

Step 4. Output the suffix block $B'_k = c_{b+1}, c_{b+2}, \ldots, c_{\mathrm{Min}(X, kN)}$, if b is not the end of block B_k, that is, if $b \ne \mathrm{Min}(X, kN)$.

As an example, let the source message contain X=2005 characters, and the block size is chosen to be N=100. The source is partitioned into Y=21 blocks of characters, where the first 20 blocks contain 100 characters each, while the last block contains 5 characters. Let the quotation q chosen be a passage in the source spanning from the 129th character to the 302nd character, that is, a=129 and b=302, as illustrated in Fig. 3. The quotation is just contained within blocks B_2, B_3, and B_4, and thus j=2 and k=4. Since j is larger than 1, we need to include the starting hash code H_1, in order for the final reader to be able to recover the subsequent hash codes H_2, H_3 and H_4. In this example, both a prefix and a suffix filling block are required., with the prefix block being $B'_2= c_{101}, c_{102}, \ldots, c_{128}$, and the suffix block being $B'_4= c_{303}, c_{304}, \ldots, c_{400}$.

The essence of Algorithm 2 is to include the filling blocks B'_j



$R = c_1, c_2, \ldots, c_{2005}$
B_1
B_2
B_3
B_4
B_5
B_6
B_7
B_{20}
B_{21}
$q = c_{129}, c_{120}, \ldots, c_{302}$
$B_2 \| B_3 \| B_4 = c_{101}, c_{102}, \ldots, c_{400}$
$B'_2 = c_{101} \ldots c_{128}$
$B'_4 = c_{303} \ldots c_{400}$

Figure 3. The example used in quotation signature generation.

and B'_k where necessary such that the blocks B_j and B_k can be reconstructed. Then using the same arguments as the case where a quotation contains full blocks of characters, the integrity of $B_j \| B_{j+1} \| \ldots \| B_k$ can be verified from the source signature S_k and the starting block-chain hash code H_{j-1} included in p. Since the concatenation contains q, the quotation is authenticated if the above verification was successful.

We note that the quotation signature contains at worst a source signature S_k, a block-chain hash code H_{j-1}, and two filling blocks of N-1 characters each, while on average, assuming uniform quotation positions a and b, N-1 characters will be included in the quotation signature for the filling blocks. The length of the quotation signature generated using the proposed method is thus linear to the block size chosen, and independent of the length of the quotation or the length of the source message. Quotation signature generation requires only simple arithmetic, so the computational cost is negligible.

The algorithm performed by the final reader to verify the integrity of an authenticated quotation is exactly the counterpart to that of Algorithm 2, and is presented below.

Algorithm 3: Quotation authentication.

Input: A quotation with texts $c_1, c_2, \ldots, c_r$, and quotation signature p that contains a source signature S, a starting block-chain hash code H^S, and filling blocks B^S and B^E, where H^S, B^S, and B^E are optional. The author's public key PK (contained in the author's digital certificate), from which the message verification function $V_{PK}(H', S)$ can be devised.

Output: The result of authentication, that is, the result of $V_{PK}(H', S)$, where H' is the reconstructed block-chain hash code calculated by the steps in the Algorithm.

Step 1. Prefix the quotation with filling block B^S and append it with the filling block B^E where applicable to obtain $c'_1, c'_2, \ldots, c'_v$. The concatenated result contains full blocks of characters, that is, v=kN, where N is the block size and k is the number of blocks. Let $B_1, B_2, \ldots, B_k$ be the blocks of characters derived by partitioning the concatenated result N characters at a time.

Step 2. If the starting block-chain hash code H^S does not exist, then the first block-chain hash code is simply the hash value of the first character block, that is, $H_1 = H(B_1)$. Otherwise, we concatenate H^S with the first block, and then calculate the hash value of the result, that is, H_1=$H(H^S \| B_1)$.

Step 3. For each integer i from 2 to k,

a. Reconstruct the subsequent block-chain hash code using the previous block-chain hash code concatenated with the current block of characters, that is, H_i=$H(H_{i-1} \| B_i)$.

Step 4. Output $V_{PK}(H_k, S)$ as the verification result.

Following the last example, where the quotation spans from the 129th character to the 302nd character in the source message, and the prefix block being B^S= $c_{101}, c_{102}, \ldots, c_{128}$, and the suffix block being B^E= $c_{303}, c_{303}, \ldots, c_{400}$. The concatenation operation in Step 1 of the above algorithm results in $c_{101}, c_{102}, \ldots, c_{400}$, and k=3. The reconstructed hash codes H_1, H_2, and H_3 are equal respectively to H_2, H_3, and H_4 in the original source.

We note that quotation authentication requires one signature verification operation, and then one hash code computation for each block to be performed. Since the computation of hash codes is much more efficient than message signing or verification operations, the computational efforts required for quotation authentication is significantly lower than that of source signature generation.

The selection of the block size N affects the computational efforts to generate the source signatures, the overall size of the source signatures, the efficiency in computing the quotation signature, the size of the quotation signature, and the computation required for quotation authentication. A larger block size requires less source signatures to be generated for the same source message, resulting in faster signature generation and smaller overall signature size. However, as pointed out previously, the average size of a quotation signature grows linearly with the block size.

Note that if we make the block size larger than or equal to the source message length, then the block signature method reduces to the whole message authentication method. If the block size is equal to one on the other extreme, then the quotation signature will be of minimal size, as there is no need for the filling blocks. However, the number of source signatures required will be large, which is undesirable both in terms of the computational efforts or the storage required.

IV. EXPERIMENTS

A series of experiments have been conducted to investigate the performance of the block signature method with different block sizes. The experiments were performed on an Acer Veriton 7700GX computer with a Pentium 4 3.4GHz CPU and 1280MB of RAM installed. The Secure Hash Algorithm 1 (SHA1) [9] is used for the one way collision-resistant hash function, and the RSA PKCS#1 standard is adopted as the message signing and verification schemes. SHA1 generates 160-bit message digests on input octets of any length, and the RSA implementation used generates 1024-bit signatures. The available implementations of the above functions in the Microsoft .NET framework works on byte arrays, so the blocks of characters must first be encoded into byte arrays. The UTF-8 encoding is used in the prototype in order to cater for messages that contain international characters. For all the experiments, the test source message contains ten thousand ASCII characters, which is approximately a two to five page document.

A. *Source Signature Generation*

In this experiment, we vary the block size and measure the time required to generate the source signatures and calculate the total size of the source signatures generated. The time required for the generation and the total size of the signatures generated are plotted in Fig. 4 and Fig.5 respectively for different block sizes. We consider only the raw signature size in the total signature size, as there are non-predictable overheads when storing the signatures inside a Microsoft Word document, making the

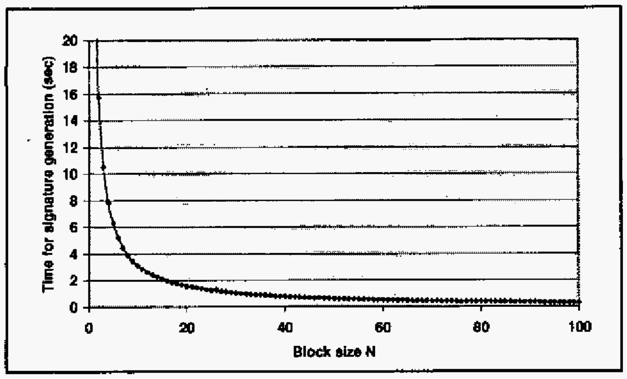


Figure 3. Time to generate block signatures of source message for different block sizes.

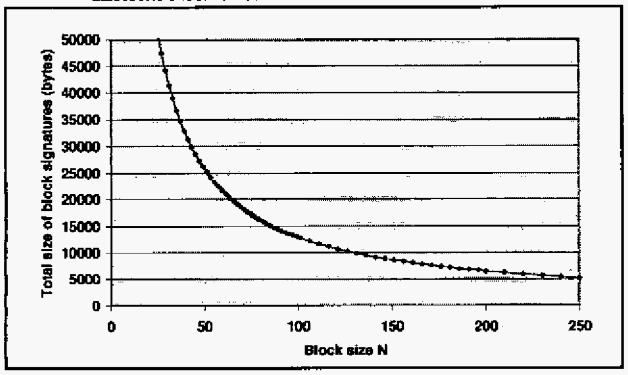


Figure 4. Total size of block signatures for different block sizes.

result misleading. The total size of source signatures is effectively the RSA signature size (1024 bit) multiplied by the number of blocks – which is equal to $128\times\lceil X/N \rceil$ bytes. We note that the block-chain hash codes can be reconstructed during quotation signature generation, and thus need not be stored in the signed source.

From Fig. 4, we observe that the source signatures can be generated in a matter of seconds for block sizes exceeding ten. The computational efforts required for signature generation is thus acceptable with sufficiently large block sizes. The storage overhead can be high compared to the source message size for smaller block sizes, as seen in Fig. 5. The total size of source signatures however decreases rapidly for increasing block size, with the total size of block signatures equal to the size of the original source when the block size equals the signature size, that is, when $N = 128$.

B. Quotation Signature and Authentication

The computational cost for quotation signature generation is negligible, so only the size of the quotation signature is of interest. In the previous section, it is determined that the average quotation signature size is N-1 characters plus the sizes of a hash value and a signature, which is N-1 + 20 + 128 = N+147 bytes in the experiment setting. For source messages that contain sentences in English, a typical quotation would contain around one to five hundred characters in length. The quotation signature in this case is approximately equal in length to the quotation for block sizes in the order of one hundred. The computational efforts required for quotation verification is significantly lower than that for source signature generation, and thus negligible by current hardware standards. A block size in the order of one hundred is recommended in view of the results of the experiments.

V. Conclusions and Future Work

In this work, a new active quotation authentication technique has been proposed to address the problem of unverified credibility by association. Traditional message authentication means cannot verify the integrity and claimed sources of quotations inside a public message. The idea of active quotation authentication is described, and includes the roles and actions of the three parties involved in this problem: the credible source, a public-message author, and the final reader.

A new block signature method has been proposed, which offers good performance for all three parties. The computational efforts required for the credible-source author is low, where messages can be signed in a matter of seconds; and the storage overhead for source signing is comparable in length to the source texts as observed by the experiments. The computational efforts for the public-message author and the final reader to respectively generate and authenticate a quotation are negligible with current-day hardware, and the storage overhead of a quotation signature is comparable in size to the quotation in question.

The proposed block signature method and the active quotation authentication technique are generic, and can be applied to other application areas. Future works can apply the techniques to other message formats such as e-mail messages and Internet web pages.

References

[1] U.S. DOE-CIAC (Computer Incident Advisory Capability) Internet Hoax Information, http://hoaxbusters.ciac.org/HBHoaxInfo.html.

[2] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication", *Advances in Cryptology - Crypto 96 Proceedings, Lecture Notes in Computer Science Vol. 1109, N. Koblitz ed,* Springer-Verlag, 1996.

[3] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: keyed-hashing for message authentication", *Internet RFC 2104*, February 1997.

[4] U.S. National Institute of Standards and Technology, NIST FIPS PUB 113, "Standard on computer data authentication", U.S. Department of Commerce, May 1985.

[5] U.S. National Institute of Standards and Technology, NIST FIPS PUB 186, " Digital signature standard (DSS)", U.S. Department of Commerce, May 1994.

[6] PKCS #1 v2.1: RSA Cryptography Standard, RSA Laboratories, June 14, 2002.

[7] Housley, R., Ford, W., Polk, W. and D. Solo "Internet X.509 public key infrastructure: X.509 certificate and CRL profile", RFC 2459, January 1999.

[8] Microsoft Office 97 Visual Basic Programmer's Guide (Microsoft Professional Editions Series). 1997.

[9] D. Eastlake and P. Jones, "US Secure Hash Algorithm 1 (SHA1)", *RFC 3174*, September 2001.

[10] W. Stallings, *Cryptography and Network Security, Principles and Practice, 2nd ed.*, Prentice-Hall, USA, 1999.