Encrypted Message Authentication by Firewalls

Chandana Gamage, Jussipekka Leiwo, and Yuliang Zheng

Peninsula School of Computing and Information Technology
Monash University, McMahons Road, Frankston, Vic 3199, Australia
{chandag,skylark,yuliang}@pscit.monash.edu.au

Abstract. Firewalls typically filter network traffic at several different layers. At application layer, filtering is based on various security relevant information encapsulated into protocol messages. The major obstacle for efficient verification of authenticity of messages at application layer is the difficulty of verifying digital signatures without disclosure of content protected by encryption. This is due to a traditional paradigm of generating a digital signature of a message and then encrypting the signature together with the message to preserve confidentiality, integrity, non-repudiation and authenticity. To overcome this limitation, a scheme shall be proposed for enabling signature verification without disclosing the content of messages. To provide maximum efficiency, the scheme is based on digital signcryption.
Keywords. Encryption, Digital Signatures, Firewalls, Confidentiality, Authenticity, Network Security, Signcryption, Public Key Cryptography

1 Introduction

Firewalls are one of the most useful and versatile tools available for securing a LAN and other applications such as constructing secure private virtual networks [21]. They are typically operated as a filtering gateway [2, 6] at the LAN-WAN interface, usually a router. Firewalls operating at data link level perform a primitive level of filtering based on frame level addressing. The network level firewalls work at a step higher and filter packets based on a set of rules including packet addresses, port addresses and possibly packet header authentication as supported by new IPv6 extensions. The most comprehensive filtering is done at the application layer with end-user level authentication of messages.

For secure communication using public key cryptography, the standard practice is for a sender to sign a message (or its hash) using her secret key and then encrypt the message *and* the signature using receivers public key. The signature is used to provide sender authenticity, message integrity and message origin non-repudiation while encryption provide message confidentiality. Other redundant information such as time-stamps or sequence numbers in messages can be used against replay and existential forgery attacks. When this cipher text message reaches its intended recipient, he first decrypts the cryptogram using his secret key. Then the signature is verified using senders public key.

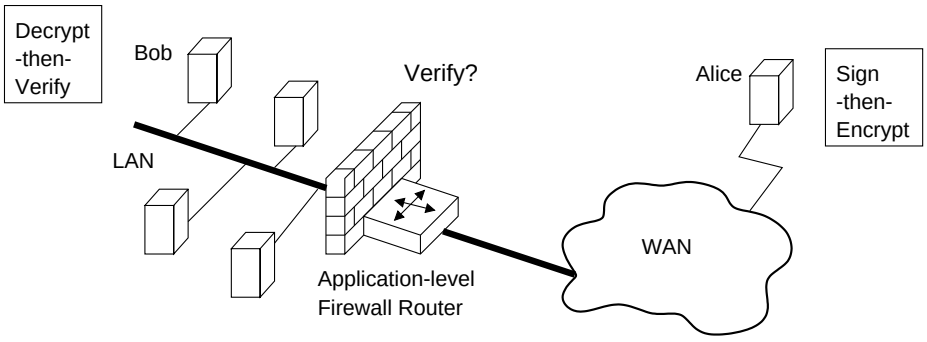

Fig. 1. Application-level firewall used for inward message authentication in a LAN

1.1 The Problem

In a LAN secured with a firewall, this standard use of public key cryptographic techniques for secure communication causes serious difficulties in filtering. As both the signed message and the signature is encrypted, the filtering process at the firewall cannot authenticate the message independent of the end-point receiver. The firewall cannot access the signature as the cryptogram cannot be decrypted without receivers secret key. This scenario is illustrated in figure 1 for communication between external user Alice and LAN user Bob.

Another problem from the users view point is that they may want to maintain the confidentiality of their communication while allowing the firewall to verify the message origin for filtering. Most widely used digital signature schemes require access to the signed text for signature verification (schemes with appendix such as DSA [17], ElGamal [11, 12] or Schnorr [27, 28]) or recover the message as part of the verification step (such as RSA [26], Rabin [25] or Nyberg-Rueppel [18, 19]).

1.2 Research Contribution

This problem of authentication of secure messages by a firewall is common to all widely used public key cryptosystems that use standard sign-then-encrypt mode of operation. We suggest that following properties should be satisfied by any practical scheme which aim to solve the problem:

Property 1. Preserve the semantics of signature-then-verification.
Property 2. Signature verified without access to the plain text.
Property 3. Should not increase the original computational and transmission costs incurred by end-user signer or verifier.
Property 4. Cost of signature verification by the firewall, measured in terms of computational and transmission effort, should not be greater than that for the end-user verifier.

In section 3 we present a complete solution to this problem which is more efficient than standard sign-then-encrypt schemes.

1.3 Structure of the Paper

There are seemingly straightforward ways to achieve authenticity without disclosure of messages in a public key cryptographic setting. These alternative mechanisms shall be summarized in section 2 and reasons pointed out why they are not capable of adequate security and objectives of this research. The proposed mechanism shall be established in section 3 and informal arguments shall be provided for security and performance of the proposal. The informal discussion shall be enhanced and a formal proof of security of the proposed scheme shall be given in section 4. Section 5 shall conclude with remarks highlighting important issues related to the proof mechanism used in this paper.

2 Related Work

We will first discuss two straightforward solutions to the problem outlined above and resulting security implications for those schemes.

Reordering If the cryptographic operations are reordered so that encryption is followed by signing, anyone can verify the signature while not compromising the confidentiality of the encrypted message. However, reordering is not a desirable option as an adversary could replace an original signature with his own in particular situations to obtain some advantage even without knowledge of the actual message content.
Chen and Hughes in [9] discuss the security protocol failures due to reordering when RSA encryption is used. Their work is an extension of the general attack presented by Anderson and Needham in [1] for protocols that sign after encryption. Apart from the apparent insecurity, this mode of operation does not satisfy the first and second properties listed earlier.

Signcryption with public key only signature verification The original signcryption primitive proposed in [30] by Zheng combines the sign-then-encrypt two-step process to create a secure authenticated message into a single logical step with significant savings in both computational and transmission costs. A disadvantage for some applications such as firewall authentication is that only the intended recipient can verify the message. A modified signcryption scheme was proposed in [3] by Bao and Deng to overcome this limitation at the cost of increased computational cost while still preserving the transmission cost savings achieved by the original scheme. Two disadvantages of this modified signcryption scheme are:

1. The signature verification only mode of operation can be used only after the original recipient has recovered the plain text message.
2. The plain text message must be forwarded to a third party for signature verification and the message confidentiality is lost.

Therefore, this scheme is unusable by a firewall as a message must be recovered by the end-user prior to firewall verification which violate the second property listed earlier. Hao Zheng and Robert Blakley [29] have also proposed a similar scheme called *Authenticryption* based on ElGamal signature

scheme and its variants. This scheme is also unusable for implementing firewall message authentication as it does not satisfy the last three properties we have stated.

3 Signcryption for Third-Party Verification

In this section, we show that with a small change to the original signcryption scheme it is possible to modify the Bao-Deng scheme to carry out signature verification without accessing the plain text. The advantages of this new mode of operation for signcryption are:

1. The cipher text only signature verification that preserves confidentiality of the original message without altering sign-then-encrypt paradigm (first and second properties).
2. The computational cost is higher than in original scheme of Zheng [30] but lower than Bao-Deng modified scheme and thus standard sign-then-encrypt schemes (third and fourth properties).
3. The transmission cost saving of the original signcryption scheme is preserved (third property).

The main parameters used in the signcryption scheme are p : a large prime number, q : a large prime factor of $p-1$, g : an integer in $[1,\ldots,p-1]$ with order $q \bmod p$, $hash$: a cryptographically strong one-way hash function of the form $\{0,1\}^* \rightarrow \{0,1\}^l$ where l is a security parameter, (E,D) : the encryption and decryption algorithms of a private key cipher such as DES, x_a : Secret key of Alice, a randomly chosen integer, y_a : Public key of Alice ($y_a = g^{x_a} \bmod p$), x_b : Secret key of Bob, a randomly chosen integer, y_b : Public key of Bob ($y_b = g^{x_b} \bmod p$) and m : a message.

3.1 Scheme for Single Prover - Single Verifier

Signcryption Choose an integer x randomly from $[1,\ldots,q-1]$ and compute $k = hash(y_b^x \bmod p)$ and $y = g^x \bmod p$. The signcrypted cryptogram (c,r,s) is computed by Alice as
$c = E_k(m)$
$r = hash(y,c)$
$s = \frac{x}{r+x_a} \bmod q$

Remark 1. We compute r by taking the hash value of c instead of m as in the original scheme. This change results in a corresponding change for the unsigncryption step. Also, we do not hash the value of y as in Bao-Deng scheme as that hashing operation is redundant. Note that we have deliberately put y before c. Here, y can be pre-computed, and hence $hash(y,c)$ can be partially pre-hashed, as every hash works in a block-by-block fashion. Otherwise if c is in front of y, then nothing can be pre-hashed until we get c.

Unsigncryption For full unsigncryption with message recovery, Bob will compute from (c, r, s)
$y = (y_a g^r)^s \bmod p$
$k = hash(y^{x_b} \bmod p)$
$m = D_k(c)$
Accept signature if and only if $hash(y, c) \stackrel{?}{=} r$

Signature Verification For partial unsigncryption with signature verification only, any verifier will compute from (c, r, s)
$y = (y_a g^r)^s \bmod p$
Accept signature if and only if $hash(y, c) \stackrel{?}{=} r$
This signature verification does not require access to the plain text message.

Use of signcryption paradigm has already satisfied our first property and the verification without message recovery shown above satisfies second property. In next section we give relative estimations of computational and transmission costs to show that third and fourth properties are also satisfied.

3.2 Discussion on Security and Performance

A question that arises due to our modification of the original signcryption scheme is whether the use of cipher text c (a public value) for computing r instead of m (a private value) weakens the resulting scheme. The value r, when viewed as corresponding to the commitment value in a three-move zero-knowledge identification scheme, only need to be a random value. For a signature scheme, this random value must also be bound to the message m. As we have used a hash function to compute r from y and c, both these conditions are satisfied. Therefore, in an *informal* analysis, the modification does not seem to reduce the security of the original signcryption scheme. However, given the major weaknesses that arise due to even minor changes to cryptographic protocols (see [1, 9]), it is essential to perform a *formal* security analysis of the proposed scheme.

Furthermore, we cannot directly use the security arguments given in the original signcryption scheme [30] as the modified schemes (both [3] and [29]) are fundamentally different due to the two step computation of the commitment value using a secret random integer. In Zheng's scheme [30], the security of the single computed value $y_b^x \bmod p$ is guaranteed by its equivalence to the *computational Diffie-Hellman problem* [10]. In Bao-Deng scheme, the computation of two values, $y_b^x \bmod p$ and $g^x \bmod p$ using the same secret random integer x does not provide such a straightforward security argument. In section 4 we give a formal proof of security based on the random oracle model [4] and show the pseudo-independence of the two computed values as an adequate guarantee of security for the signature scheme.

In digital signature generation and verification, the computational effort is dominated by the exponentiation modulo p. Other computational costs due to modular multiplication, addition, inversion and also hashing and symmetric key encryption constitute only a small fraction of the overall cost. Therefore, when

Table 1. Comparison of number of exponentiations modulo p

Operation	Signcryption	Modified Signcryption	DSA sign + ElGamal encrypt
Signcrypt	1 EXP	2 EXP	$1 + 2$ EXP
Unsigncrypt	2 EXP (1.17)	3 EXP (2.17)	$1 + 2$ EXP $(1 + 1.17)$
Verify only	n/a	2 EXP (1.17)	n/a

we try to improve the performance of digital signature schemes, the main aim is to reduce the number of modular exponentiations in the scheme. In table 1 we show that Bao-Deng scheme modified by us can verify a signature at the cost of 4 modular exponentiations as against 5 for the original Bao-Deng method. The values within parenthesis show the instances where 2 modular exponentiations can be done for the cost of 1.17 modular exponentiations using the algorithm for simultaneous multiple exponentiations [16, page 618]. In table 2 we show that the modified signcryption scheme in signature verification only mode can achieve nearly a 40% saving in computational cost over a standard DSA-ElGamal style scheme for secure and authenticated message transmission.

Table 2. Computational cost savings for modified signcryption over DSA-ElGamal

Operating mode of the modified scheme	Cost saving
Signcryption with message recovery	5/6 (4.17/5.17) 17% (19%)
Signcryption with verification only	4/6 (3.17/5.17) 33% (39%)

4 Formal Proof of Security for Verification only Mode

The security of a cryptographic protocol such as an encryption scheme or a signature scheme can be informally established through its resistance to cryptanalytic attacks. However, a more desirable guarantee of security is a formal proof that provides arguments for the strength of a particular scheme in a given computational model. Currently, there are two main techniques to achieve this goal of provable security: (1) *complexity theoretic arguments* that provide computational reductions to well-known presumably hard problems such as the discrete logarithm problem, the RSA problem, Diffie-Hellman problem, etc. (2) *random oracle technique* described by Bellare and Rogaway [4] which provide a new paradigm for security analysis through replacement of hash functions in protocols by an ideally random oracle.

To analyze the security of the verification only signcryption mode, we apply the security arguments developed by Pointcheval and Stern for digital signature

schemes [23, 22, 24] using random oracle technique of Bellare and Rogaway. The main result of Pointcheval and Stern is the *Forking Lemma* which gives a probability of finding a forking pair of signatures in the random oracle model giving an asymptotic reduction to a hard problem.

4.1 Security of a Digital Signature Scheme

There are two main classes of attacks on digital signature schemes and we will briefly describe the attacks and their consequences based on the definitions by Goldwasser, Micali and Rivest [15]:

1. *Key only* or *no message* attacks in which an attacker $\mathcal{A}$ has access only to public parameters and public keys.
2. Message attacks in which $\mathcal{A}$ has access to pairs of message texts and corresponding signatures. These known message attacks can be further categorized to four modes depending on the power $\mathcal{A}$ has on selecting messages signed by the legitimate signer Σ.
   (a) *Known-messages* in which $\mathcal{A}$ does not choose messages signed by Σ.
   (b) *Generic chosen-messages* in which $\mathcal{A}$ choose a set of messages to be signed before knowing the actual Σ targeted for attack.
   (c) *Directed chosen-messages* in which $\mathcal{A}$ choose a set of messages to be signed after selecting a specific Σ but before the actual attack.
   (d) *Adaptive chosen-messages* in which $\mathcal{A}$ choose messages for signing dynamically after inspecting signatures he obtained for previous messages.

The no message attack is the weakest type of attack on a digital signature scheme while the adaptive chosen-message attack is the strongest. The outcome of attacks on signature schemes are forgeries. There are four main types of forgeries:

1. *Total break* in which $\mathcal{A}$ recovers the secret key of Σ under attack.
2. *Universal forgery* in which $\mathcal{A}$ does not obtain the secret key of Σ but gains the ability to generate valid signatures for any message.
3. *Selective forgery* in which $\mathcal{A}$ does not obtain the secret key of Σ but gains the ability to generate valid signatures for any set of preselected messages.
4. *Existential forgery* in which $\mathcal{A}$ is able to create at least one new message and signature pair without knowing the secret key. However, the messages are only arbitrary bit strings and $\mathcal{A}$ does not have any power over their composition.

The total break is the hardest type of forgery to make while existential forgery is the easiest type of subversion of a digital signature scheme.

Therefore, a proverbly secure digital signature schemes is defined as one that could withstand an adaptive chosen-message attack (strongest) to create an existential forgery (easiest). Here the attacker is assumed to run in probabilistic polynomial time and the success of a forgery to have a non-negligible probability. The attacker $\mathcal{A}$, oracle $\mathcal{O}$ and signer Σ are all modeled as probabilistic

polynomial time Turing machines in the security analysis to follow. The chosen message attack is modeled by allowing $\mathcal{A}$ to query Σ as an oracle. We summarize the discussion on digital signature security in the random oracle model with the following two definitions.

Definition 1. *A signature scheme is (T, Q, ϵ)-secure if an attacker $\mathcal{A}$ who is limited to Q queries from the random oracle $\mathcal{O}$ over a period of time T can create an existentially forged signature with probability at most ϵ after a no-message attack. The probability is taken over the coin flips of $\mathcal{A}$ and $\mathcal{O}$.*

Definition 2. *A signature scheme is (T, Q, R, ϵ)-secure if an attacker $\mathcal{A}$ who is limited to Q queries from the random oracle $\mathcal{O}$ and R queries from the signing oracle Σ over a period of time T can create an existentially forged signature with probability at most ϵ after a chosen-message attack. The probability is taken over the coin flips of $\mathcal{A}$, $\mathcal{O}$ and Σ.*

4.2 Signature Schemes from ZK Identification Schemes

Fiat and Shamir in [14] have described a three-move identification protocol between a prover and a verifier that is perfect zero-knowledge against an honest-verifier. They have also used a general technique to derive a provably secure signature scheme from the ZK identification protocol and an improved version of this signature scheme was presented by Feige, Fiat and Shamir in [13] which we recall below.

The setup phase of the signature scheme chooses two distinct primes p and q randomly and compute the composite integer $n = pq$. The two primes p and q are kept secret while n is the public modulus. For a security parameter k which is a positive integer, distinct integers $s_1, \ldots, s_k \in \mathbb{Z}_n^*$ are chosen randomly. A public key K_p which is a tuple $(v_1, \ldots, v_k)$ is computed as $v_j = s_j^{-2} \bmod n$, $1 \leq j \leq k$ and the corresponding private key K_s is the tuple $(s_1, \ldots, s_k)$. The scheme uses a one-way hash function $hash : \{0,1\}^* \rightarrow \{0,1\}^k$ where the security parameter k is chosen to prevent off-line attacks on the hash function.

1. Prover chooses a random value (commitment) r, $1 \leq r \leq n-1$, and compute the value (witness) $u = r^2 \bmod n$.
2. Prover computes the random value (challenge) $e = (e_1, \ldots, e_k)$ where each $e_i \in \{0,1\}$ as $e = hash(m||u)$ for a message $m \in \{0,1\}^*$.
3. Prover computes the value (response) $s = r \cdot \prod_{j=1}^{k} s_j^{e_j} \bmod n$.
4. Prover sends the signature (e, s) and message m to verifier.
5. Verifier computes the value $w = s^2 \cdot \prod_{j=1}^{k} v_j^{e_j} \bmod n$ and $e' = hash(m||w)$. The signature is accepted if and only if $e' = e$. This step is the signature verification test.

Remark 2. We make following observation on the necessary attributes of signature schemes that belong to the class derived from ZK identification protocols. The transmitted signed message consists of the tuple (challenge, response, message), where:

1. The witness value is a random permutation from a very large set.
2. The challenge is simply a one-way hash of the message being signed and the witness value.
3. The response is bound only to witness, challenge, message and private key K_s.

4.3 Properties of Modified Signcryption Scheme

Drawing from the above observations, we now show that the signature verification only mode has the necessary attributes that make the modified scheme to be within the class of signatures derived from ZK identification schemes.

1. The commitment value is the random integer x and the witness value is y. If the length of the output of hash function is sufficiently large, then y is a random permutation from a large set of size $\lceil \log_2 p \rceil$ for a given x.
2. The challenge r is a one-way hash of the cipher text c and the witness y. As our intention is to authenticate the cipher message at the input to the firewall, use of c instead of the plain text m does not affect the security of the scheme.
3. The response is computed from commitment x (therefore, equivalently the witness), challenge r (therefore, including the cipher message c) and private key of signer x_a.

4.4 Security Results

Arguments for a (T, Q, ϵ)-secure Scheme. We assume a no message attack by $\mathcal{A}$ with access to $\mathcal{O}$ and public key of Σ with security parameter l. If $\mathcal{A}$ is successful in an existential forgery within a time bound T and random oracle query bound Q with probability of success $\epsilon \geq 7Q/2^l$, then the Forking Lemma of Pointcheval and Stern [24, Theorem 10] states that DLP in sub groups of prime order can be solved in expected time less than $84480QT/\epsilon$.

The proof of above claim can be directly shown by using the same approach in [24] for the Schnorr signature scheme: After a polynomial replay of $\mathcal{A}$, we obtain two valid signatures, (c, r, s) from signing oracle σ and (c, r', s') from random oracle $\mathcal{O}$ with $r \neq r'$, for the same cipher message using modified signcryption scheme. Then we have the following two equalities as part of the signature verification test: $y = (y_a g^r)^s \bmod p$ and $y = (y_a g^{r'})^{s'} \bmod p$. By solving the two equations we can compute the secret key x_a of Σ as $\log_g y_a = \frac{(rs - r's')}{(s' - s)} \bmod q$. That is, if a signature can be successfully forged for any message then the DLP can be efficiently solved to reveal the secret values. It is important to note that the reduction is to the basic discrete logarithm problem although the security of the signcryption scheme is based on computational Diffie-Hellman problem which is argued to be less secure [7].

Arguments for a (T, Q, R, ϵ)-secure Scheme. We assume an adaptive chosen-message attack by $\mathcal{A}$ with access to $\mathcal{O}$ and public key of Σ with security parameter l. Furthermore $\mathcal{A}$ can query Σ as an oracle. If $\mathcal{A}$ is successful in an existential forgery within a time bound T, random oracle query bound Q and signing oracle query bound R with probability of success $\epsilon \geq 10(R+1)(R+Q)/2^l$, then the Forking Lemma of Pointcheval and Stern [24, Theorem 13] states that DLP in sub groups of prime order can be solved in expected time less than $120686QT/\epsilon$.

Similar to the proof in the original paper we only need to show that two signatures can be forked without using the secret value of Σ. This is done by showing the signatures σ due to Σ and signatures σ' due to $\mathcal{O}$ have the same probability distribution.

$$\sigma = \left\{ (c, r, s) \;\middle|\; \begin{array}{l} x \in_R (\mathbb{Z}/q\mathbb{Z})^* \\ k = hash(y_b^x \bmod p) \\ y = g^x \bmod p \\ c = E_k(m) \\ r = hash(y, c) \\ s = x/(r + x_a) \bmod q \end{array} \right\} \text{ and } \sigma' = \left\{ (c, r, s) \;\middle|\; \begin{array}{l} x \in_R \mathbb{Z}/q\mathbb{Z} \\ r \in_R \mathbb{Z}/q\mathbb{Z} \\ s = x \\ c \in \{0,1\}^* \\ t = (y_a g^r)^s \bmod p \\ y = hash(t) \\ t \neq 1 \bmod p \end{array} \right\}$$

The probabilities of obtaining a signature σ with r computed by Σ and σ' with r obtained from $\mathcal{O}$ such that $y = hash((y_a g^r)^s \bmod p) \neq 1 \bmod p$ are

$$\Pr_{\sigma}[c, r, s] = \Pr_{x \neq 0, r}[c, r, s] = \frac{1}{(q-1)2^l} \text{ and } \Pr_{\sigma'}[c, r, s] = \Pr_{y, r}[c, r, s] = \frac{1}{(q-1)2^l}$$

Finally, if Σ chooses the integer x uniformly and randomly, then the two values $t = y_b^x \bmod p$ and $y = g^x \bmod p$ are (pseudo) independent as both g and $y_b = g^{x_b} \bmod p$ are generators in $\mathbb{Z}_p^*$ of order q where q is a prime. This ensures that the signature verification and partial recovery of bits at the firewall does not leak information that can be used in an attack on breaking message confidentiality or signature forgery.

5 Conclusions

The security proof given in section 4 provide only an asymptotic security analysis (compared to the notion of exact security [5]). However, it is possible to give the exact security of the proposed scheme using the concrete security analysis methodology of Ohta and Okamoto [20] based on the ID reduction technique.

As a concluding remark, we observe that Canetti, Goldreich and Halevi [8] have given counter-examples for protocols proverbly secure in the random oracle model but found to be insecure in practical implantation using cryptographic hash functions. More importantly, the specific counter-example they have provided, *correlation intractability*, is at the core of the three-move ZK identification scheme to signature scheme conversion technique of Fiat-Shamir that we have

used for constructing our proof. However, as yet we have not found any security weaknesses in the proposed scheme for authentication of encrypted messages by a network firewall due to the findings in [8].

References

[1] R. Anderson and R. Needham. Robustness principles for public key protocols. In D. Coppersmith, editor, *Advances in Cryptology - CRYPTO'95*, volume 963 of *Lecture Notes in Computer Science*, pages 236–247. Springer-Verlag, 1995.

[2] F. M. Avolio and M. J. Ranum. A network perimeter with secure external access. In *Proceedings of the* 3^{rd} *Annual System Administration, Networking and Security Conference (SANS III)*, pages 1–14. Open Systems Conference Board, 1994.

[3] F. Bao and R. H. Deng. A signcryption scheme with signature directly verifiable by public key. In H. Imai and Y. Zheng, editors, *Public Key Cryptography - PKC'98*, volume 1431 of *Lecture Notes in Computer Science*, pages 55–59. Springer-Verlag, 1998.

[4] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the* 1^{st} *ACM Conference on Computer and Communications Security*, pages 62–73. ACM Press, 1993.

[5] M. Bellare and P. Rogaway. The exact security of digital signatures - how to sign with RSA and Rabin. In U. M. Maurer, editor, *Advances in Cryptology - EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416. Springer-Verlag, 1996.

[6] S. M. Bellowin and W. R. Cheswick. *Firewalls and Internet Security.* Addison-Wesley, 1994.

[7] D. Boneh. The decision Diffie-Hellman problem. In x, editor, *Proceedings of the* 3^{rd} *Algorithmic Number Theory Symposium*, volume 1423 of *Lecture Notes in Computer Science*, pages 48–63. Springer-Verlag, 1998.

[8] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. In *Proceedings of the* 30^{th} *Annual ACM Symposium on Theory of Computing*, pages 9–9. ACM Press, 1998. (to appear).

[9] M. Chen and E. Hughes. Protocol failures related to order of encryption and signature - computation of discrete logarithms in RSA groups. In C. Boyd and E. Dawson, editors, *Information Security and Privacy - ACISP'98*, volume 1438 of *Lecture Notes in Computer Science*, pages 238–249. Springer-Verlag, 1998.

[10] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, November 1976.

[11] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and D. Chaum, editors, *Advances in Cryptology - CRYPTO'84*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer-Verlag, 1985.

[12] T. ElGamal. A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, IT-31(4):469–472, July 1985.

[13] U. Feige, A. Fiat, and A. Shamir. Zero-knowledge proofs of identity. *Journal of Cryptology*, 1:77–94, 1988.

[14] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *Advances in Cryptology - CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer-Verlag, 1987.

[15] S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal of Computing*, 17(2):281–308, April 1988.

[16] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[17] National Institute of Standards and Technology, U.S. Department of Commerce. *Digital Signature Standard. Federal Information Processing Standards Publication (FIPS PUB) 186*, 1994.

[18] K. Nyberg and R. A. Rueppel. A new signature scheme based on the DSA giving message recovery. In *Proceedings of the* 1^{st} *ACM Conference on Computer and Communications Security*, pages 58–61. ACM Press, 1993.

[19] K. Nyberg and R. A. Rueppel. Message recovery for signature schemes based on the discrete logarithm problem. *Designs, Codes and Cryptography*, 7:61–81, 1996.

[20] K. Ohta and T. Okamoto. On concrete security treatment of signatures derived from identification. In H. Krawczyk, editor, *Advances in Cryptology - CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 354–369. Springer-Verlag, 1998.

[21] R. Oppliger. Internet security: Firewalls and beyond. *Communications of the ACM*, 40(5):92–102, May 1997.

[22] D. Pointcheval and J. Stern. Provably secure blind signature schemes. In U. M. Maurer, editor, *Advances in Cryptology - ASIACRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 387–398. Springer-Verlag, 1996.

[23] D. Pointcheval and J. Stern. Security proofs for signature schemes. In U. M. Maurer, editor, *Advances in Cryptology - EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 387–398. Springer-Verlag, 1996.

[24] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 9:9–9, 1999.

[25] M. O. Rabin. Digitalized signatures and public key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, MIT Laboratory for Computer Science, January 1979.

[26] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

[27] C.-P. Schnorr. Efficient identification and signatures for smart cards. In G. Brassard, editor, *Advances in Cryptology - CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252. Springer-Verlag, 1990.

[28] C.-P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[29] H. Zheng and G. R. Blakley. Authenticryption: Secrecy with authentication. Manuscript, 1998.

[30] Y. Zheng. Digital signcryption or how to achieve cost(signature & encryption) $\ll$ cost(signature) + cost(encryption). In B. S. Kaliski, editor, *Advances in Cryptology - CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 165–179. Springer-Verlag, 1997.