# Certified Email: A Survey

Brian Curtis†
Australian Catholic University

Jan Seruga‡
Australian Catholic University

**Abstract**

A comparison between various proposals for the provision of certified email is presented. In order to place the proposals in context, an outline of the basic principles of secure email is provided, as far as possible separating the issues of confidentiality, integrity and authenticity.

**Keywords:** email, cryptography, internet protocols.

## 1. Introduction

Certified mail is frequently used in the "real world" to provide to the sender, a receipt certifying the delivery of an item of mail to its intended recipient. Basically the postal service requires a signature from the recipient prior to handing over the mail item and delivers this back to the sender as proof that the transaction has been successfully completed. Oppliger [1] has pointed out that an electronic equivalent is a missing piece of the infrastructure required for professional use of email.

An implicit assumption in what follows is that the Public Key Infrastructure (PKI) is capable of reliably providing identification of the parties registered within it. This assumption is somewhat naïve as has been pointed out in [2] and in [3] with particular reference to the Certificate Authorities lack of checking connected with online applications for class I certificates.

It is also the case that not all participants in email transactions can or want to register in PKI. For both these reasons methods to limit the requirement of PKI participation will be considered.

## 2. Notation

The following notation will be consistently applied.

| | |
|---|---|
| *S, R* | The sender S and recipient R of a message. Also used as their "identity". |
| *A -> B* | *A* sends to *B* via a non-secure channel (e.g. email). |
| *A => B* | *A* sends to *B* via a secure channel (e.g. SSL). |
| *A*c, *A*p | The private (Confidential) and public (Published) keys of *A*. |
| **E**( k, m ) | The asymmetric Encryption of the message m using the key k. Here we assume: $m = \mathbf{E}(A\text{c}, \mathbf{E}(A\text{p}, m)) = \mathbf{E}(A\text{p}, \mathbf{E}(A\text{c}, m))$ |
| **C**( k, m ) | The symmetric enCryption of the clear message m using the key k. |
| **D**( k, e ) | The symmetric Decryption of the encrypted message e using the key k. We simply assume: $m = \mathbf{D}(k, \mathbf{C}(k, m))$ |
| **H**( m ) | A digest (Hash) of the message m. |
| [m \| n] | The unambiguous concatenation of message components m and n. |

Note that as there is no specific notation given for digital signatures. Although a digital signature is not "Encryption with a private key", the obverse does hold for RSA and simplifies the discussion somewhat.

## 3. Ensuring Message Integrity and Authenticity

S signs the message with his private key giving:

$s = \mathbf{E}(S\text{c}, \mathbf{H}(m))$

*S->R* $sm = [\, m \mid s \,]$

The authenticity of *S* is guaranteed as only the public key *S*p can decrypt s giving **H**(m). In turn, **H**(m) guarantees the integrity of the message. Consequently, *S* cannot repudiate m.

In principle, *S* could simply publish *S*p so guaranteeing integrity, but authenticity requires *S* register in PKI so *R* can confidently retrieve *S*p.

The use of PKI could be reduced by the use of the ISP of *S, I*.

*S->I* $[\, \mathbf{H}(m) \mid S \,]$

*I* can authenticate *S* by use of a pre-established password.

*I->S* s = **E**( *I*c, [ **H**(m) | *S* ] )
*S->R* sm = [ m | s ]

S can then authenticate *I* via PKI and consequently *S*.

Note that the communication and computation burden on *I* is small and independent of the size of m. Further, no trust is placed in *I* by *S* as the confidentiality of m is not compromised and *S* can ensure the validity of s by checking that:

m ?= **E**( *I*p, s ).

However *I* could fraudulently originate an email apparently signed by *S*.

This is essentially a "poor-man's" PKI with added complications.

## 4. Ensuring Message Confidentiality

*S->R* em = **E**( *R*p, m )

Confidentiality is assured as only *R* can decrypt em using *R*c . This method implies nothing about the authenticity of *S*. Similarly the integrity of m is only weakly assured by the intelligibility of the decryption of em.

A much more efficient scheme using a session key k, randomly generated by S, is:

*S->R* em = [ **C**( k, m ) | **E**( *R*p, k ) ]

These schemes require *R* register in PKI to ensure *S* does not choose a fraudulent *R*p.

Again, PKI use could be limited by use of the recipients ISP, *J*.

*S->R* em = [ **C**( k, m ) | **E**( Jp, [ k | *R* ] ) ]
*R->J* **E**( *J*p, [ k | *R* ] )

*J* can authenticate *R* by use of a pre-established secret and decrypt:

[ k | *R* ] = **E**( Jc, **E**( *J*p, [ k | *R* ] ) )
*J->R* k

From which R can decrypt:

m = **D**( k, **C**( k, m ) )

Note that while *J* could breach the confidentiality of m simply by intercepting em, no other interceptor could gain k without *R*'s password for *J*.

This scheme will be further considered below with regard to certified email.

## 5. Ensuring Integrity, Authenticity and Confidentiality

*S->R* sem = [ **C**( k, m ) | **E**( *R*p, k ) | **E**( *S*c, **H**(m) ) ]

Here both *S* and *R* must register in PKI for reasons given above.

Established email methods are used for the transmission and the two parties need never be online simultaneously. The additional communication overhead is simply the acquisition by *S* of *R*p and, on receipt, the acquisition of *S*p by *R*.

The only trust placed in a third party is the assumption that PKI can reliably provide authentic public keys for *S* and *R*.

A weakening of the need for PKI can be provided by methods given above, with a slight increase in protocol complexity and some increase in the possibility of fraud.

## 6. Confirming Message Receipt

In the above discussion implicit faith is placed in the email transmission system to actually deliver the message. If this delivery is to be assured to the sender (or proved in court), some form of certified email method is required. The requirement is for a receipt certificate that the sender can use to prove delivery of the email independent of the honesty or diligence of the recipient. Such a certificate should be available to the sender if and only if the intended recipient actually received the email. Note that this requirement removes from consideration schemes that provide only for a recipient to optionally generate a receipt.

Consider the following naïve protocol using a digital signature as a receipt certificate:

*S -> R* m
*R -> S* r = **E**( *R*c, **H**(m) )

The certificate, r, can obviously be used by S to prove receipt of m by R. However the protocol is subject to abuse should R fail to send (or S fail to receive) r – for whatever reason. What is needed is the ability to require R to generate a receipt as a precondition for reading the email. A number of schemes have been proposed and are outlined below. The problem is considered as orthogonal to the integrity and confidentiality of the transmission although any proposed mechanism may integrate one or both of these aspects.

### 7. Use of Oblivious Transfer

The general perception of email is one of very limited interaction. A user composes an email, initiates its transmission and, once an (possibly temporary) online connection is established, the email is sent and no further interaction takes place. Similarly, at a later time, the recipient simply establishes their connection and receives one or more emails before possibly disconnecting. Thus there is no need for both parties to be simultaneously online. We see this property as fundamental to the concept of email and a property that distinguishes it from an online conversation or message exchange mechanism.

With the added functionality of certification, it is not unreasonable to expect a small number of extra rounds of communication. However, schemes such as [4] that require the exchange of a message encryption key on a bit by bit basis using oblivious transfer, while of interest, either generate an inordinate number of communication rounds, or, if implemented synchronously, abrogate the above fundamental property.

### 8. Use of a Light Trusted 3rd Party [5]

This protocol assumes the existence of a third party *T*, with whom *R* is registered, that is willing to provide a simple decryption and receipt generation service. The following is a simplified outline.

*S* generates a random key k and symmetrically encrypts the message:

| | |
|---|---|
| | em = **C**( k, m ) |
| *S*->*R* | [ em \| **E**( *T*p, [ k \| *R* \| S \| **H**(em) ] ) ] |

*R* now generates her own hash:

| | |
|---|---|
| | h = **H**(em) |
| *R*=>*T* | [ **E**( *T*p, [ k \| *R* \| S \| **H**(em) ] ) \| h ] |

*T* authenticates *R* by means of a shared secret (e.g. password) and compares:

h ?= **H**(em)

to determine if *S* & *R* are bound to the same message. If not, the protocol terminates. Otherwise it "simultaneously" sends two messages:

| | | |
|---|---|---|
| *T*->*S* | receipt = **E**( *T*c, [ h \| *R* ] ) | |
| *T*=>*R* | k | (presumable back over the secure channel already established) |

*R* can now decode the message:

m = **D**( k, em )

A potential flaw in this protocol is that the "simultaneous" transmission by T of the receipt and the key may experience catastrophic problems causing one of the items to fail to be received. If that party is R then it should be a simple matter for her to request retransmission. If the party is S, he would have no way of determining if the message was received or read and the protocol would be broken. He could retransmit the original message but would be relying on the good faith of R to recapitulate her part of the protocol. In principle he could request that T check that the protocol has been completed and, if so, retransmit the receipt. This does not appear to be directly catered for.

Since m is only ever transmitted encrypted as em, and clear-text k is only sent over a secure channel, the confidentiality of m "falls out" of the protocol - unless T maliciously intercepts and decodes it. Similarly the integrity of em is assured by the use of the digest **H**(em)

This protocol is effectively an augmentation of the above version of confidentiality [section 4.] (where PKI use is limited), with the addition of an integrity check, using h, and the requirement on T to send a receipt certificate back to S.

A practical issue with this protocol is that the registration with T is of *R* rather than *S*. This would be reasonable in cases where *R* expects to receive many certified emails but the situation where *S* wishes to send many certified emails is at least equally likely. In fact the paper does allow the protocol to specify no authorization for *R*, but this weakens the validity of the certificate substantially and, in particular allows *S* to cheat.

### 9. Use of a Public Forum [6]

This protocol makes use of a "time stamped forum" whose contents are publicly available (e.g. The New York Times or a Usenet newsgroup).

*S* generates a random key k to symmetrically encrypt the message:

| | |
|---|---|
| *S*->*R* | em = **C**( k, m ) |

*R* signs a request for *S* to publish the key k in a public Forum by a given Date:

| | |
|---|---|
| *R*->*S* | rcpt = **E**( *R*c, [ **H**(em) \| Date \| Forum ] ) |

Later:

| | |
|---|---|
| *S*: | publishes [ **H**(em) \| k ] in Forum by Date |
| *R*: | retrieves the key and decrypts the message |

Note that rcpt represents a conditional receipt of m that achieves full validity by virtue of the publication in the forum. An external arbiter may then confirm that the published digest **H**(em) matches the request of *R* and decode the message with the publish key k – which of course was presumably available to *R* from the forum. If the publication was incorrect or not timely, the arbiter would deny the receipts validity.

The authors propose an "optimistic" extension to this protocol that makes the actual publication step unnecessary should *R* "play fair". After *S* receives rcpt, instead of publishing, he simply sends *R* the key:

*S->R* [ **H**(em) | k ]
*R->S* frcpt = **E**( *R*c, [ **H**(em) | k ] )

If *R* does not so reply to S within the timeframe specified in rcpt, *S* continues with the original protocol and publishes.

The authors note that the confidentiality of the message may be breached by an interception of both em and rcpt as anyone can decode rcpt and lookup the forum to find k.

Clearly, the integrity of the message is assured as the digest in *R*'s request must match that in the publication. While the authenticity of *R* is assured by her signature in rcpt, no guarantee of the identity of *S* is provided except as a return email address. Obviously *R*'s registration with PKI is mandatory for rcpt to be binding.

Although an interesting idea the protocol does involve a number of rounds of communication between presumably off-line participants. The original email must be answered by *R* (with rcpt) and only then may *S* must send the key allowing *R* to actually read the email. If the optimistic assumption of *R*'s fairness does not materialize, *S* must then publish in some forum on penalty of repudiation by *R*.

## 10. Use of Offline Trusted 3rd Party [7]

Nanadic et al [7] propose an optimistic protocol based on the use of a cryptographic primitive called a "verifiable and recoverable encrypted signature" (VRES). This enables the recipient to encrypt his receipt in a manner in which the sender can verify the correctness of the receipt without accessing its content while assuring the sender that an agreed TC can recover the receipt from this encryption should the recipient refuse to do so (optimistically, a rare occurrence). In the optimistic case the protocol requires two rounds of communication, firstly for an extended hash of the message and the return of a VRES and secondly for the message itself and the return of an actual receipt. The TC may be invoked by the sender to recover the receipt, but only on provision of the actual message, which the TC must also forward to the recipient. Although trust is only occasionally placed in the TC the opportunity for its corrupt or erroneous behavior is still present.

## 11. Further Work

In the above the gamut of existing solutions to the problem of email certification were examined and their practical limitations and vulnerabilities discussed. Our further work focuses on the incorporation of secure audit mechanisms to mitigate vulnerabilities present in these schemes.

## References

[1] R. Oppliger, "Certified Mail: The Next Challenge for Secure Messaging", *Commun. ACM*, vol. 47, no. 8, pp. 75-79, Aug. 2004.

[2] C. Ellison, and B. Schneier, "Inside Risks: Risks of PKI: Secure Email", *Commun. ACM,* vol. 43, no. 1, pp. 160, Jan. 2000.

[3] A. Levi, and Ç. K. Koç, "Inside Risks: Risks in Email Security", *Commun. ACM,* vol. 44, no. 8, pp. 112, Aug. 2001.

[4] S. Even, O. Goldreich, and A. Lempel, 1985. "A randomized protocol for signing contracts", *Commun. ACM,* vol. 28, no. 6, pp. 637-647, Jun. 1985.

[5] M. Abadi, N. Glew, B. Horne, and B. Pinkas, "Certified Email with a Light On-line Trusted Third Party: Design and Implementation", in *Proceedings of the Eleventh International World Wide Web Conference,* May 2002, pp. 387-395.

[6] B. Schneier, and J. Riordan, "A Certified E-mail Protocol", in *Proceedings of 13th Annual Computer Security Applications Conference,* Dec. 1998, pp. 100-106.

[7] A. Nenadić, N. Zhang, and S. Barton, "Fair certified e-mail delivery", in *Proceedings of the 2004 ACM Symposium on Applied Computing*, Mar. 2004, pp. 391-396.

† School of Business and Informatics,
Nth. Sydney, N.S.W. 2060
b.curtis@mackillop.acu.edu.au

‡ School of Business and Informatics,
Nth. Sydney, N.S.W. 2060
j.seruga@mackillop.acu.edu.au