# Posta Elettronica e sicurezza: una applicazione

## Electronic mail and security: a case study

*Francesco Gennai, Marina Buzzi*

*Istituto per le Applicazioni Telematiche, CNR - Pisa, Italy*

### *Abstract*

*La firma digitale associata a messaggi di Posta Elettronica fornisce l'autenticazione del mittente, l'integrità del messaggio e il non ripudio dell'origine.*
*L'operazione di verifica viene usualmente effettuata dai client di Posta Elettronica, a vantaggio dell'utente. Comunque, in particolari contesti, dove i messaggi in ingresso sono soggetti a processi di elaborazione automatica del contenuto, la verifica della firma digitale via client di Posta Elettronica potrebbe diventare un "collo di bottiglia" (bottleneck), degradando le prestazioni del sistema.*
*In questo lavoro viene descritta la progettazione e realizzazione di un software per l'automazione del processo di verifica della firma digitale associata a messaggi di Posta Elettronica. Il sistema è attualmente utilizzato in una sperimentazione intesa a semplificare la gestione delle richieste di registrazione di nomi a dominio sotto il Top Level Domain .it.*

*Digital signature can be associated with Internet messages in order to guarantee authentication, message integrity and non-repudiation of origin. Verification of incoming digitally signed messages is usually implemented by e-mail clients (receiving agents) implementing this function on behalf of the end-user. However in specific context, where message content is automatically elaborated, signature verification by e-mail client could become a system bottleneck, thus justifying an automatic verification system.*
*In this work we briefly describe our experience in designing and implementing software in order to automate the verification process of signed e-mail. The system has been designed to simplify the registration of Internet domains under the .IT Top Level Domain.*

## 1. Introduzione

Il processo di verifica della firma digitale associata a messaggi di Posta Elettronica (PE) viene usualmente effettuato da un client di PE, a vantaggio dell'utente. Se il processo non ha successo, o se ci sono delle condizioni di warning (per es. l'indirizzo del campo from non corrisponde ad uno degli indirizzi presenti nel certificato) il client allerta l'utente.
Il destinatario gioca comunque un ruolo fondamentale nell'intero processo di verifica in quanto:

- è capace di correlare il contenuto del messaggio con il mittente, attribuendogli quindi un significato;
- autorizza il client a caricare e/o rimuovere i certificati di Certification Authority di cui si fida (trusted CA) e a recuperare la Certificate Revocation List (CRL) delle CA stesse, affinchè il processo di verifica possa avvenire correttamente.

D'altra parte, l'elaborazione automatica del contenuto di messaggi, velocizza l'elaborazione dei dati e riduce gli errori umani. In tale contesto, la verifica della firma digitale via client di PE risulterebbe onerosa in termini di tempi e di risorse umane e, al crescere del numero dei messaggi, potrebbe diventare un collo di bottiglia rallentando le prestazioni dell'intero sistema.
Per capire come un processo automatico può sostituirsi ad un processo in parte interattivo, possiamo fare due considerazioni importanti che si ricollegano alle due precedenti:

- in questo scenario il contenuto del messaggio è sostanzialmente un modulo elettronico (form) contenente campi prefissati conformi ad una certa sintassi che il processo automatico è in grado di riconoscere e interpretare, e quindi non esiste un legame semantico specifico tra il contenuto del messaggio e il mittente;
- le funzioni tramite le quali l'utente determina le Certification Authority di fiducia (trusted CA) o schedula il caricamento delle CRL devono essere trasferite ad un amministratore del sistema automatico, in modo da mantenere il controllo sul processo di verifica.

Il nostro studio prevede sia la progettazione e sviluppo di un sistema automatizzato per la verifica della firma digitale associata a messaggi, sia l'attivazione di meccanismi di sicurezza lato server che consentano di svolgere questo servizio con un appropriato grado di "sicurezza" ed "affidabilità" delle operazioni.

## 2. Lo studio

Il primo passo per la progettazione del sistema è stato studiare i meccanismi con cui la tecnologia a Chiave Pubblica (Public Key Technology) si integra con i sistemi di PE e capire come interpretare i servizi di sicurezza basati su crittografia, ed in particolar modo come verificare messaggi digitalmente firmati. A tale scopo occorre capire come:

- riconoscere le parti MIME del messaggio contenenti i dati protetti e le protezioni,
- applicare correttamente il processo di verifica dei dati protetti sulla base dei valori estratti.

In relazione al primo aspetto, L'RFC 1847: "Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted" [Gal 95] specifica come applicare servizi di sicurezza alle parti MIME del messaggio. Esso aggiunge due nuovi "Content type" (Multipart/Signed e Multipart/Encrypted) costituiti da due sezioni: una per i dati protetti e l'altra per le informazioni di controllo necessarie per rimuovere la protezione.
Questo RFC rappresenta la base su cui è stato costruito la specifica S/MIME (Secure MIME), costituita nella sua versione consolidata (v. 3) da una suite di RFC (2630 [Hou 99], 2631[Res 99], 2632 [Ram 99a], 2633 [Ram 99b] , 2634 [Hof 99]) che definiscono sia il formato dei dati, sia i meccanismi applicabili.
In particolare, l'RFC 2630 descrive la sintassi (Cryptographic Message Syntax) utilizzata per apporre la firma digitale, per creare un digest, cifrare un messaggio, etc. mentre l'RFC 2633 definisce i tipi MIME application/pkcs7-mime e application/pkcs7-signature, utilizzati per trasportare messaggi S/MIME firmati.
Il secondo punto da comprendere è la corretta applicazione del processo di verifica alle parti MIME estratte. Questa parte include i meccanismi per il recupero del certificato e la sua validazione. L'RFC 2632 specifica le regole base che i client (receiving agent) devono applicare per verificare in modo corretto un messaggio firmato in ingresso al sistema.
In aggiunta, l'Internet Draft (I-D) "Internet X.509 Public Key Infrastructure Certificate and CRL Profile" [Hou 01] (che è parte di una famiglia di specifiche per la Public Key Infrastructure (PKI) X.509 per Internet) descrive il formato e la semantica del certificato e della Certificate Revocation List (CRL) e le

procedure per l'elaborazione dei cammini di certificazione (certification path) in ambiente Internet. Questo I-D fornisce una strutturazione di base in cui gestire certificati e CRL.
Infine il toolkit di pubblico dominio OpenSSL [AA.VV. 01b] è stato fondamentale per l'implementazione del sistema automatico di verifica dei messaggi ("Message Verify" system). Il pacchetto offre delle librerie per lo sviluppo di applicativi crittografici e degli applicativi per la manipolazione/generazione/conversione di vari oggetti appartenenti ad una PKI (certificate manipulation, basic CRL manipulation, basic CA management, signing, encrypting, decrypting, verifying, etc…). Grazie a questo software l'implementazione del sistema è risultata semplice e al tempo stesso robusta.

## 3. Il sistema

L'operazione di validazione della firma è una operazione piuttosto critica affidata all'intervento manuale dell'interlocutore (via client di posta elettronica). Vista però come "chiave" per l'autorizzazione all'utilizzo di un certo servizio, allora il concetto di validazione della firma può essere spostato su sistemi automatici capaci di processare un elevato numero di messaggi nell'unità di tempo.
Nel nostro contesto, la verifica di un messaggio è utilizzata come chiave di accesso per la registrazione di nomi a dominio (*domain names)*. Ogni messaggio infatti contiene un form di registrazione per un dominio. Contenuti differenti, se ricevuti, sono rifiutati. Se il processo di verifica ha successo, la richiesta è accettata ed elaborata, altrimenti è rigettata ed una notifica automatica viene inviata al mittente.
In passi con cui è possibile schematizzare l'intero processo sono:

a) L'utente invia il messaggio firmato a un mailbox configurato per l'elaborazione automatica del contenuto del messaggio;
b) All'ingresso del sistema al messaggio viene assegnato un identificatore univoco;
c) Viene verificata la firma;
d) Se la firma è verificata con successo, solo la parte firmata è estratta e passata alle successive fasi;
e) Se la verifica fallisce con un errore temporaneo (p. es. CRL non disponibile) il messaggio viene accodato per successivi tentativi (fino al raggiungimento di una soglia prefissata).
f) Se la verifica fallisce con un errore permanente (per es. messaggio non firmato), viene inviata una notifica di errore al mittente e all'amministratore locale.

Il sistema "Message Verify" (MV) è totalmente configurabile e monitorabile via interfacce web. La figura 1. mostra lo schema logico del sistema.
I certificati delle CA di fiducia (trusted CAs) possono essere aggiunte al (o rimosse dal) *CA Database* dall'amministratore del sistema. Naturalmente il sistema realizzato è in grado di utilizzare certificati emessi da qualsiasi CA, previo caricamento del certificato/i della CA root ed eventuali subCA nel database del sistema.
Le CRL sono automaticamente scaricate (utilizzando il campo Certificate's *CRL Distribution Point*) dal processo *CRL Manager* che aggiorna il database delle CRL locale (*CRL Database*). LDAP query sono naturalmente possibili, ma non ancora implementate.
Il *Message database* include anche richieste ricevute via fax, memorizzate come file postscript (*Message/Fax Database*). Un identificatore globale viene assegnato ad ogni messaggio o fax che entra nel sistema, mantenendo la sequenza temporale delle richieste; questo è veramente importante per risolvere collisioni su richieste dello stesso nome a dominio.

Il *Message Status Cache* è utilizzato per migliorare l'efficienza ed anche per mantenere separate le informazioni relative ai processi da quelle dei messaggi/fax: esso mantiene informazioni temporanee sullo stato del messaggio, durante il suo tempo di vita all'interno del sistema.
Il processo di verifica (*Verification Process)* aggiunge quattro campi all'header del messaggio, che specificano rispettivamente: il *Distinguished Name* di chi ha emesso il certificato (*Certificate Issuer Distinguished Name)*, il *Distinguished Name* del possessore del certificato *(Certificate Subject Distinguished Name),* il numero di serie del certificato (*Certificate Serial Number*) e la concatenazione tra il codice di ritorno del *Verification Process* e l'identificatore globale del messaggio (X-MVcertissuer:, X-MVcertsubject:, X-MVserialnumber, X-MVglobalid:). In tal modo il messaggio archiviato mantiene informazioni sull'esito dell'operazione di verifica.

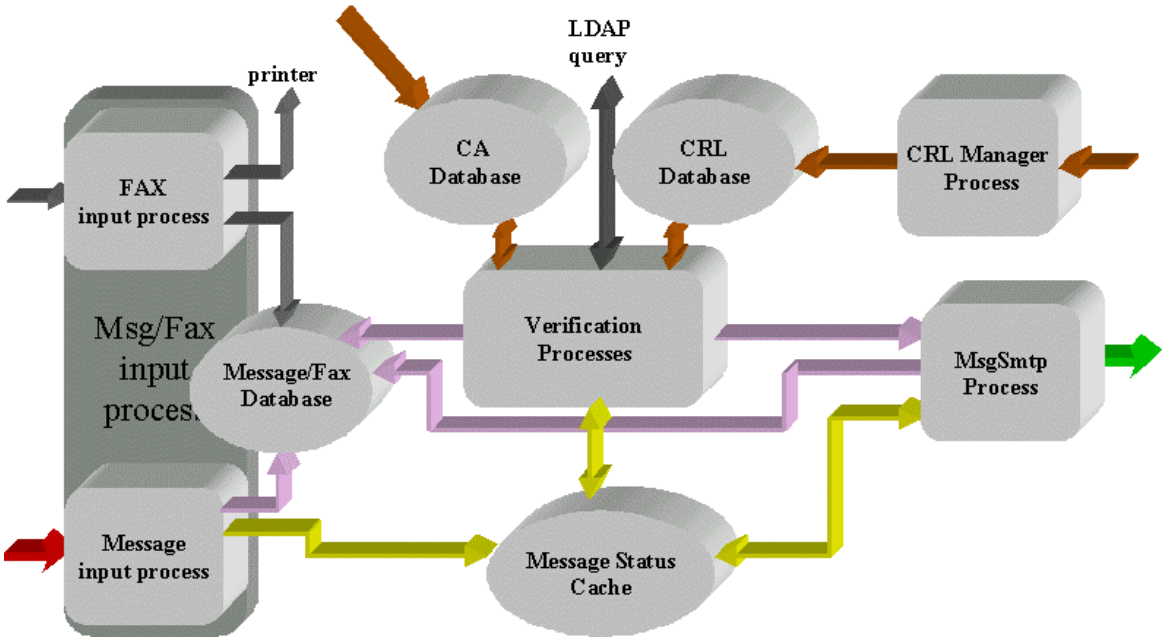


**Fig.1 - Schema logico del sistema Message Verify**

Questi valori sono anche utili per gli operatori (di help desk) che utilizzano interfacce web per accedere al Msg/Fax database.
Infine, poichè i messaggi sono processati in parallelo su un cluster di 2-nodi, dopo l'elaborazione potrebbero trovarsi fuori sequenza. Il processo *MsgSmtp* ri-ordina i messaggi nella corretta sequenza temporale, rispettando lo stato dei messaggi riportato nel *Message Status Cache*, prima di inviarli al processo di elaborazione del contenuto.

## 4. Notifiche ed errori

Il processo di verifica può fallire se:
- L'indirizzo del from non corrisponde ad uno degli indirizzi presenti nel certificato;
- Non esiste un valido percorso fino ad una CA affidabile;

- E' impossibile accedere ad una CRL;
- La CRL non è valida;
- La CRL è scaduta;
- Il certificato è scaduto;
- Il certificato è stato revocato.

Il fallimento può essere temporaneo o permanente. Per condizione di errore temporanee (per es. impossibile accedere ad una CRL) il sistema prevede un certo numero di successivi tentativi, fino al raggiungimento di una soglia prestabilita, configurabile.
Notifiche ed errori sono segnalati nel campo Subject dell'header del messaggio e in altri campi nel corpo del messaggio (From:, Global ID:, Message ID:, Previous Message ID:, Errortype:, Number of retries:, Cert issuer: e Cert subject:).

## 5. Interfacce

La prima pagina dell'interfaccia richiede un accesso controllato da user/password. Sono consentiti due tipi di accesso (figura 2):
- normale, per gli operatori - consente il browsing del database messaggi/fax;
- privilegiato, per l'amministratore - consente di accedere alle funzioni di configurazione.


**Fig. 2 - Interfaccia del sistema Message Verify**

In particolare l'accesso privilegiato consente di:
- aggiungere o rimuovere certificati di CA;
- gestire il database dei certificati delle CA, ed in particolare visualizzare i campi presenti nei certificati;
- visualizzare il log del processo di gestione delle CRL in formato compatto o esteso;

- configurare il sistema;
- monitorare il sistema (processi).

La figura 3 mostra l'interfaccia di accesso al database fax/messaggi acceduta in modo sequenziale. Naturalmente sono possibili ricerche per data o per identificatore univoco del msg/fax.

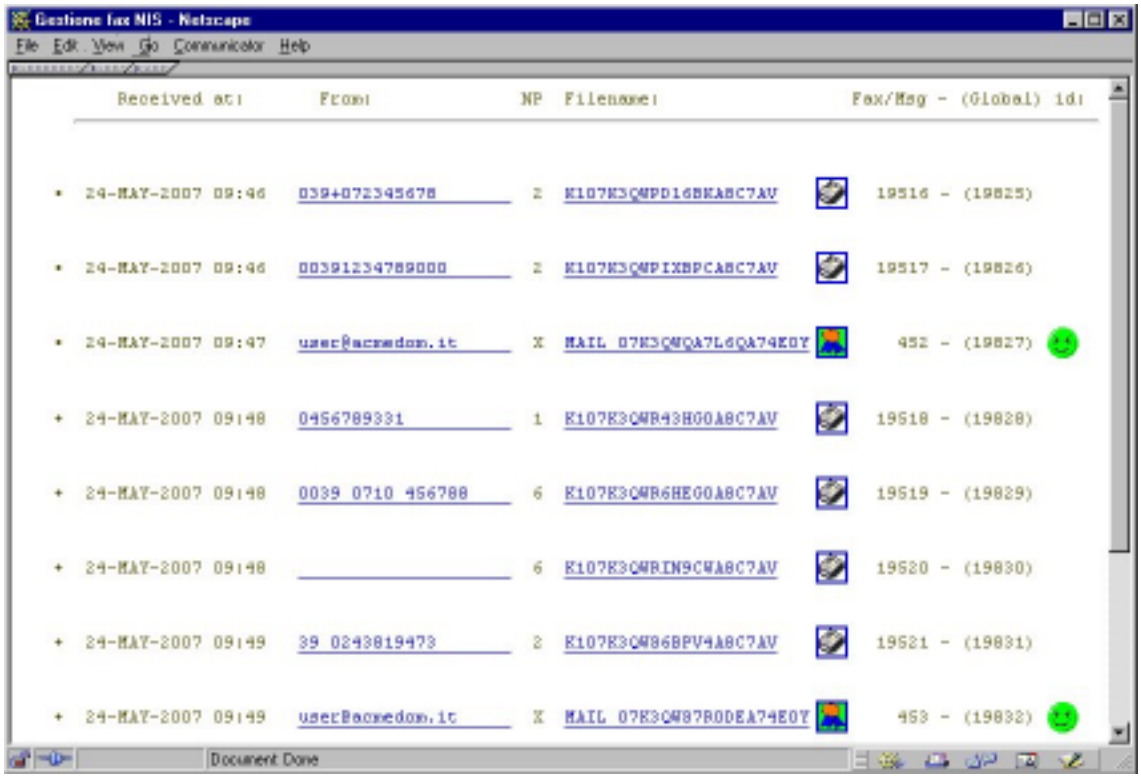


**Fig. 3 - Browsing del database messaggi/fax**

La figura 4 mostra una porzione dell'interfaccia di configurazione del sistema. Come si può vedere dalla figura il sistema è stato reso il più possibile configurabile, in modo da adattarsi velocemente a cambiamenti nella policy dell'organizzazione.

## 6. Performance

L'ambiente operativo è un cluster OpenVMS costituito da 2 nodi Alphaserver 800 (500 Mhz).
Per verificare le performance del sistema, prima della sua messa in funzione, è stato effettuato un test effettuato sull'invio di 2000 messaggi firmati. I risultati sono mostrati nella tabella 1.

| | 6 Job (3 x nodo) | 1 Job | 2 Job (1 x nodo) |
|---|---|---|---|
| Processo verifica | 45 min | 2h 6min | 59 min |
| Processo delivery | 1h 16 min | 2h 6min | 1h 4 min |
| *Totale* | *1h 16 min* | *2h 6min* | *1h 4 min* |

**Tab. 1 - Tempi di esecuzione del test**

Questi risultati evidenziano come le migliori prestazioni si ottengano attivando un singolo job su ognuno dei due nodi del cluster.

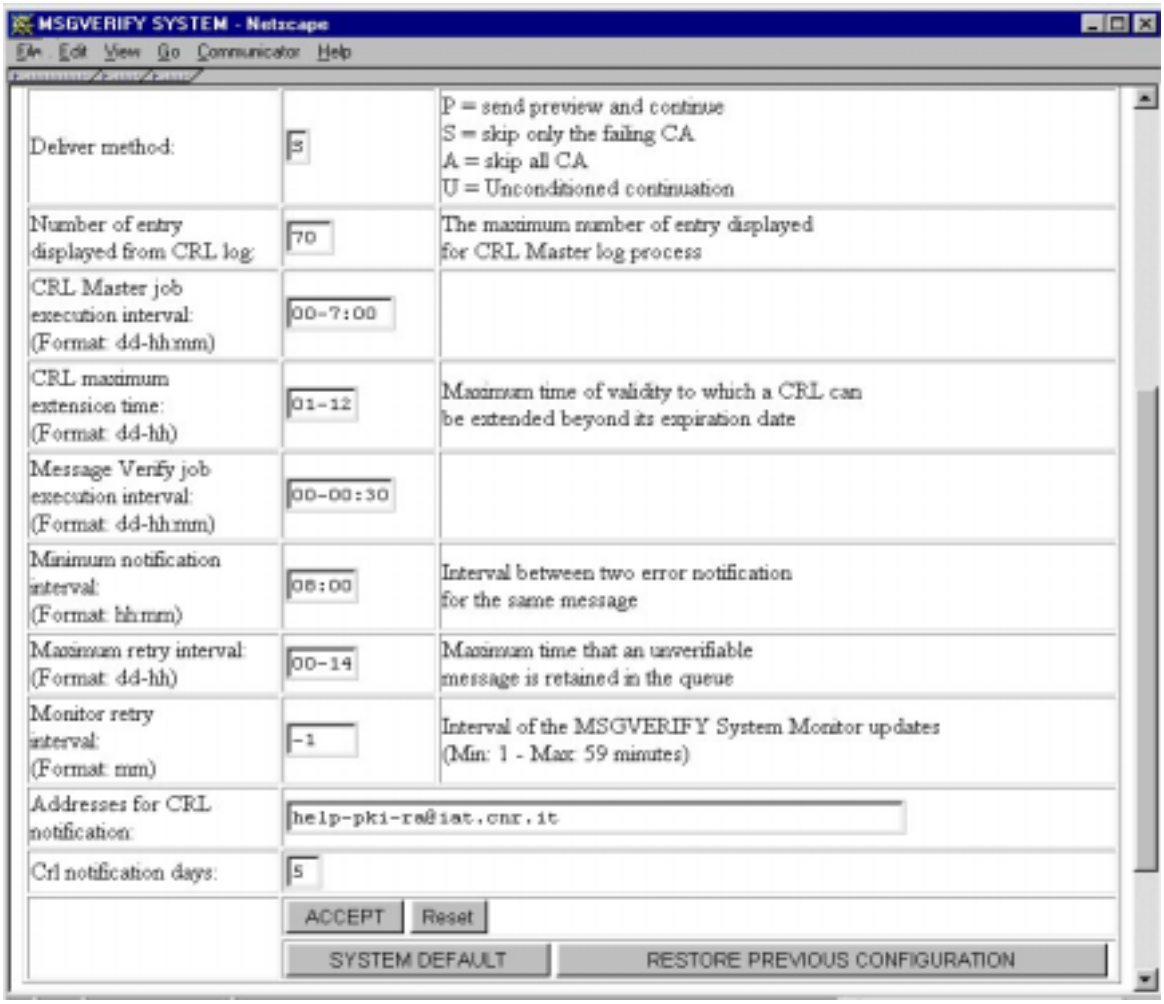


**Fig. 4 - Interfaccia di configurazione del sistema Message Verify**

## 7. Conclusioni

Il sistema è stato sviluppato per permettere la verifica automatica della firma digitale di un flusso di messaggi destinati a ulteriori elaborazioni automatiche. Il contesto di sviluppo e sperimentazione del sistema è stata la Registration Authority italiana che riceve circa 25000 richieste di registrazione di domini per mese [AA.VV. 01a].
La sperimentazione del sistema, ha manifestato numerosi vantaggi:

- Efficienza nel servizio;

- Miglioramento nella qualità del servizio: la firma digitale fornisce un metodo di autenticazione forte, rispetto ai tradizionali meccanismi di user e password (autenticazione debole).
- Trasparenza del servizio per gli operatori - non è richiesta alcuna formazione;
- Affidabilità nel processo di verifica della firma digitale e nella archiviazione dei dati;
- Basso costo.

Il sistema è attivo da alcuni mesi e non si sono rilevati problemi. Questo lavoro è in evoluzione dato che il sistema è mantenuto aggiornato con eventuali nuove proposte e raccomandazioni internazionali.

## Bibliografia


**[AA.VV. 01a]** The Italian Registration Authority - http://www.nic.it/.

**[AA.VV. 01b]** The OpenSSL Project - http://www.openssl.org/.

**[Gal 95]** J. Galvin, S. Murphy, S. Crocker N. Freed - RFC 1847. Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted. http://www.imc.org/rfc1847, October 1995.

**[Hof 99]** P. Hoffman. RFC 2634: Enhanced Security Services for S/MIME, http://www.ietf.org/rfc/rfc2634.txt, June 1999.

**[Hou 99]** R. Housley. RFC 2630: Cryptographic Message Syntax, http://www.imc.org/rfc2630, June 1999.

**[Hou 01]** Housley,W. Ford, W. Polk, D. Solo - I-D: Internet X.509 Public Key Infrastructure Certificate and CRL Profile. http://www.ietf.org/internet-drafts/draft-ietf-pkix-new-part1-06.txt, April, 2001.

**[Ram 99a]** B. Ramsdell. RFC 2632: S/MIME Version 3 Certificate Handling. http://www.ietf.org/rfc/rfc2632.txt, June1999.

**[Ram 99b]** B. Ramsdell - RFC 2633: S/MIME Version 3 Message Specification. http://www.ietf.org/rfc/rfc2633.txt, June 1999.

**[Res 99]** E. Rescorla. RFC 2631: Diffie-Hellman Key Agreement Method, http://www.ietf.org/rfc/rfc2631.txt, June 1999.