

Secure Electronic Data Interchange over the Internet

The Electronic Data Interchange over the Internet (EDI-INT) standards provide a secure means of transporting EDI and XML business documents over the Internet. EDI-INT includes different implementation protocols that work over the Internet's three major transports — SMTP, HTTP, and FTP. Each uses Secure Multipurpose Internet Mail Extensions (S/MIME), digital signatures, encryption, and message-receipt validation to ensure the necessary security for business-to-business communications.

Kyle Meadors
Drummond Group

Numerous retailers, manufacturers, and other companies within business supply chains are leveraging Applicability Statement #2 (AS2) and other standards developed by the IETF's Electronic Data Interchange over the Internet (EDI-INT) working group (www.imc.org/ietf-ediint/). Founded in 1996 to develop a secure transport service for EDI business documents, the EDI-INT WG later expanded its focus to include XML and virtually any other electronic business-documentation format. It began by providing the digital security and message-receipt validation for Internet communication for MIME (Multipurpose Internet Mail Extensions) packaging of EDI.¹

EDI-INT has since become the leading means of business-to-business (B2B) transport for retail and other industries.

Although invisible to the consumer, standards for secure electronic communication of purchase orders, invoices, and other business transactions are helping enterprises drive down costs and offer flexibility in B2B relationships. EDI-INT provides digital security of email, Web, and FTP payloads through authentication, content-integrity, confidentiality, and receipt validation.

History

Western Union first demonstrated EDI in the 1850s with the telegraph, but lack of infrastructure hindered its wider use. By the 1960s, however, the necessary technology was available to facilitate EDI among business trading partners. As EDI took hold and replaced hardcopy delivery, large companies, such as Sears and General Motors, created their own pro-

Glossary

This list includes the major terms and acronyms from the world of EDI-INT.

- AS1: Applicability Statement #1 was the EDI-INT working group's first standard; it is designed to work over email (SMTP).
- AS2: Applicability Statement #2 is the most popular EDI-INT applicability standard; it is designed to work over the Web (HTTP).
- AS3: Applicability Statement #3 is the latest EDI-INT applicability standard; it is designed to work over FTP.
- EDI: Electronic Data Interchange is a standardized format for interorganizational business-data exchanges.
- Edifact: EDI for Administration, Commerce, and Transport, a standard developed by the UN, is the most popular EDI standard in Europe and Asia.
- EDI-INT: EDI over the Internet is an open standard that provides data security and message validation.
- MDN: A message disposition notification notifies the originator in an EDI-INT exchange that its trading partner has received the EDI-INT message.
- MIC: The message-integrity check (also called a digest) is the hash value of the EDI payload document.
- PKCS: Public-key cryptography standards use public-private key pairs for encryption and digital-signature creation.
- NRR: Nonrepudiation of receipt is a legal event that ensures that a receiving trading partner can't deny having received a specific message.
- SHA-1: Secure hash algorithm 1 (pronounced "shah one") is a hash, or one-way, encryption algorithm developed by the US National Institute of Standards and Technology (NIST).
- S/MIME: Secure MIME uses PKCS standards with all messages to add security to basic MIME.
- TLS: The Transport-Layer Security protocol is used to encrypt the transmission of an HTTP message.
- VANs: Value-added networks use proprietary security mechanisms to let trading partners exchange EDI messages.
- X.509: This standard, originally developed within the International Telecommunications Union and now supported by the IETF, is widely used for creating and processing digital certificates.
- X12: This EDI standard, developed by the American National Standards Institute (ANSI), is the most popular in North America.
- 3DES: The Triple Data-Encryption Standard (pronounced "triple-des") is a quick and powerful symmetric encryption algorithm for data confidentiality.

proprietary formats – some smaller companies even developed forms of EDI for internal accounting.

Despite improving processing efficiency, these proprietary solutions prevented interoperability between business partners, unless companies supported each partner's EDI formats. This lack of interoperability necessitated additional software, human translation, or a return to hardcopy exchange. To solve this dilemma, several organizations developed open standards for EDI – the two most popular of which were X12 and EDI for Administration, Commerce, and Transport (Edifact). Currently in its fifth version since the American National Standards Institute released it in 1982, X12 is the most common EDI standard for North America. Edifact, a truly international standard, has become the general choice in Europe, Asia, and most of the world since the United Nations sponsored its release in 1985. Although other EDI standards exist, most people use the generic term EDI to describe business documents stored in these or other open standards.

Value-Added Networks

The goal behind EDI is to model traditional business transactions in a standard digital format. *Data elements* define individual pieces of information

within documents, such as trading partner names or item costs. Each EDI standard defines numerous *transaction sets*, which are sequences of data elements that reflect specific types of business transaction. In X12, for example, transaction set number 850 represents purchase orders. Receipt of an 850 transaction set thus informs a company that its trading partner wishes to purchase something; by parsing the set, the company can find out the necessary information to complete the purchase.

Although Edifact and X12 define EDI data formats, they provide no standard way to transport the electronic documents. From the 1960s through the 1990s, businesses primarily used proprietary communication networks, known as value-added networks (VANs), to send EDI data to trading partners. VANs used leased communication lines and provided added-value services, such as security and receipt confirmation, which were essential for secure business transactions. Because they were proprietary, however, VANs weren't readily interoperable. If a business used one VAN and its trading partner used another, they couldn't trade electronically unless they paid extra costs to get their VANs to communicate. These added service charges tended to prevent smaller trading partners from participating.

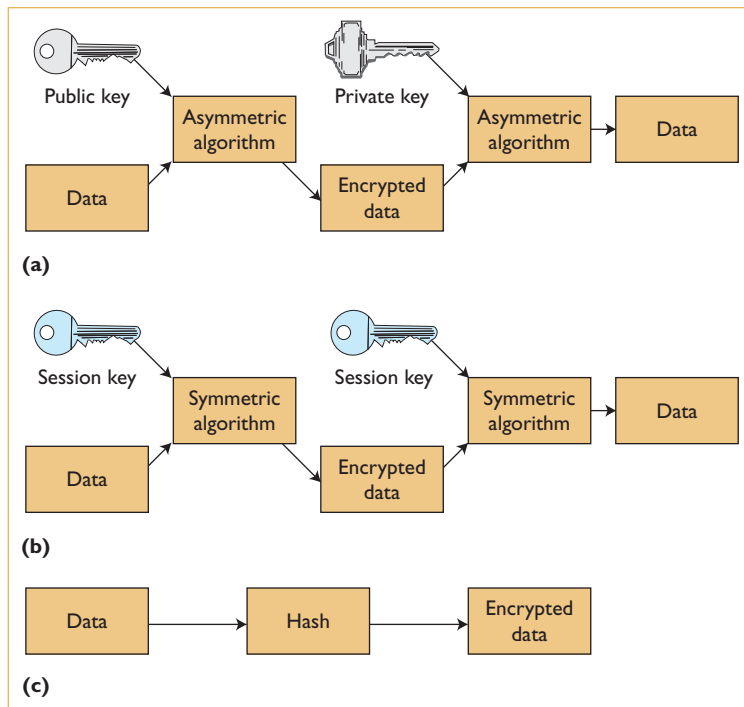


Figure 1. Encryption techniques. (a) Asymmetric encryption (or public-key cryptography) requires separate public and private keys for encryption and decryption. (b) Symmetric encryption (or session-key cryptography) uses a single “secret” key for both encryption and decryption. (c) Rather than using a key, hash (or one-way) encryption creates a unique “fingerprint” for data. Hash-encrypted data is not recoverable.

Internet-Based EDI

Leveraging the Internet’s widespread acceptance in the 1990s offered the potential to achieve an alternative to VANs. Yet, even as the Internet provides the benefits of openness and flexibility, it also presents liabilities in terms of security and reliability, which are essential for business transactions.

With the development and acceptance of X12 and Edifact, the IETF’s EDI working group began working to define a MIME-encapsulation² approach for EDI. By defining how to store an EDI document in an Internet message, enterprises could transmit EDI transactions over the Internet, albeit without security. As XML’s popularity as an electronic business document format grew, the IETF Network WG developed a MIME format for it as well.³ The EDI-INT WG then began building on this foundation to provide security and message validation for effective EDI and XML transport. EDI-INT security should provide mechanisms that

- let message recipients authenticate originators,
- notify originators of message receipt,

- validate document content integrity, and
- ensure confidentiality for the intended recipient.

EDI-INT would then define applicability standards for utilizing these security goals in exchanging message documents over the three major Internet protocols – SMTP, HTTP, and FTP.

EDI-INT Technology

The current work on EDI-INT data security builds on Secure MIME (S/MIME) formats,⁴ public-key cryptography standards (PKCS), and X.509 digital certificates. Given the early work of encapsulating EDI payloads in MIME, S/MIME is well suited for EDI-INT. A mature standard that’s widely implemented within security toolkit software, S/MIME adds digital signatures and data encryption to MIME messages. It derives its security features from PKCS encryption algorithms and formats.⁵ PKCS encryption is considered secure as long as accompanying digital certificates use key values of 1,024 bits. Even with future advances in computation, larger key lengths such as 2,048 bits will still provide sufficient protection.

A digital certificate contains information about its owner (subject), issuer, validity period, and associated public key.⁶ The issuer is generally a certificate authority, such as Entrust or VeriSign, which has trustworthy policies and procedures to ensure that the certificate accurately represents the owner and that the public-private key pair (which enables asymmetric encryption) is valid and unique. Yet, several EDI-INT applications provide certificate managers that create self-signed certificates, in which the subject and issuer are the same. This can be beneficial when trading partners are willing to extend trust to the operator that generated the certificate. Every EDI-INT application employs one or more X.509 standard certificates for its data encryption and digital signatures. Implementers can use either a single digital certificate for both or separate certificates for the two features.

The certificate owner stores the private key safely within the application and provides the digital certificate, with the public key, to its trading partner. Only the certificate owner knows the private key, and there’s no risk of security compromise from anyone else knowing the public key alone.

Asymmetric encryption of PKCS works on the mathematical model that one key can encrypt information that only the other key can decrypt (see Figure 1a). For example, the PKCS encryption

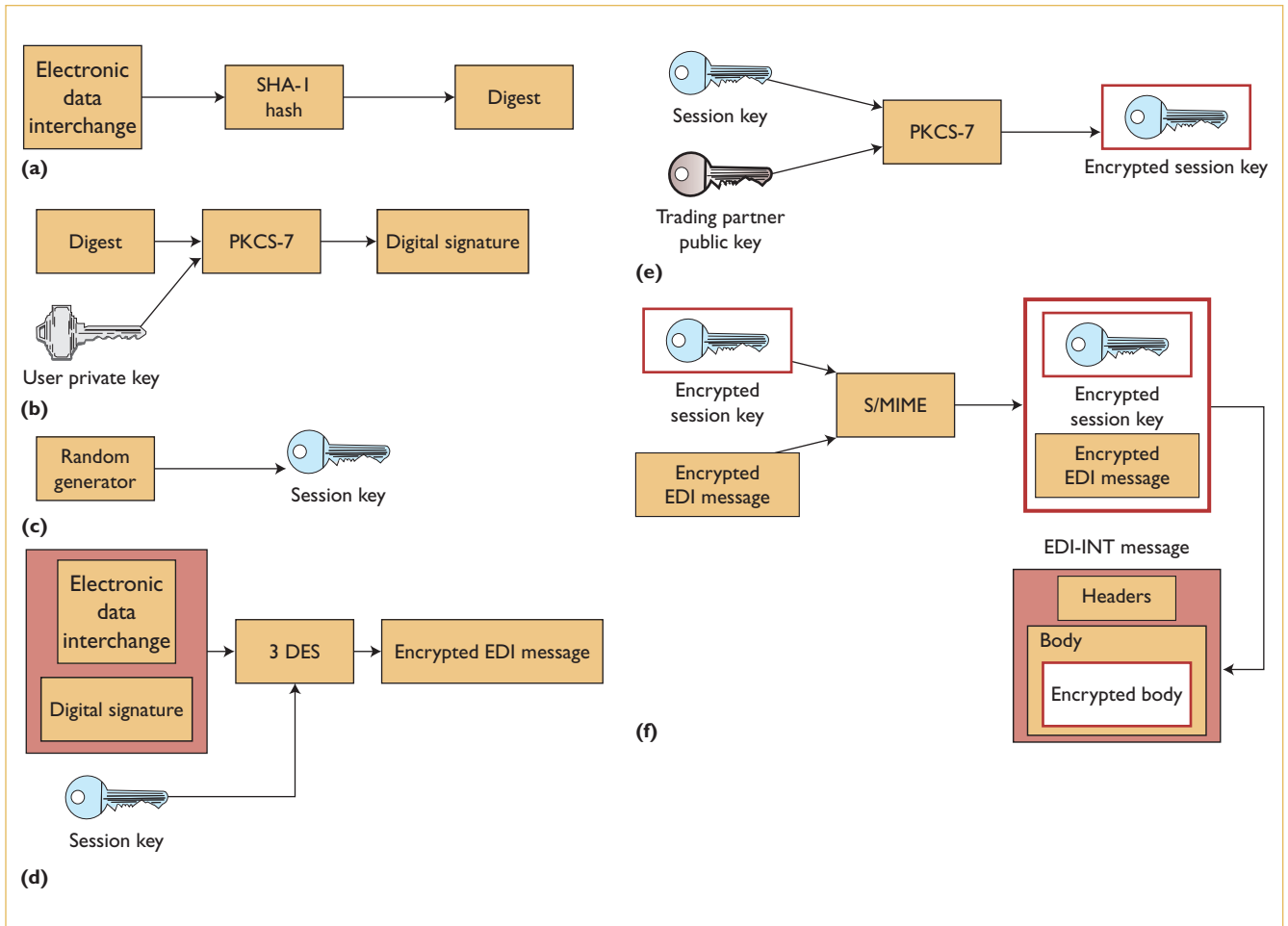


Figure 2. Creating digital signatures and encryption. (a) To compute a message digest, the secure hash algorithm 1 (SHA-1) creates a unique hash of the EDI message. (b) To create a digital signature, the hash and user’s private key are run through the PKCS algorithm to create an encrypted version of the message digest. (c) To generate random session keys, a simple algorithm chooses a random number of substantial size (for example, 56 bits). (d) To encrypt EDI messages, the secret session key and EDI message run through a symmetric algorithm, such the Triple Data Encryption Standard (3DES). (e) To encrypt session keys, the PKCS asymmetric algorithm uses the session key and the recipient’s public key to create an encrypted session key. (f) To construct the S/MIME envelope structure, the user agent stores the encrypted EDI message and encrypted session key in an S/MIME structure encoded in Abstract Syntax Notation number one (ASN.1).

algorithm takes two inputs – the data to be encrypted and the public key – to produce the encrypted output. The decryption process uses the same algorithm, but the encrypted output is paired with the private key as inputs to resolve the original payload data. As a result, information encrypted using the public key can be decrypted only with the private key. Equally important, asymmetric encryption algorithms can be used with either the public or private key. If data is asymmetrically encrypted with a user’s private key, then only that user’s public key can decrypt the data. This has important implications when applying asymmetric encryption within data security.

Application of Data Security

Using PKCS security technology and S/MIME formatting, EDI-INT can provide digital signatures to ensure that transferred business documents are identical to the originals (content integrity) and to verify the originator’s identity (authentication).

Digital Signatures and Encryption

Figure 2 illustrates the steps for creating digital signatures and encryption.

To create a digital signature, the EDI-INT application first runs a payload document through secure hash algorithm 1 (SHA-1), which produces a unique digest of the document (see Figure 1c).

Hashes such as SHA-1 are one-way encryption functions that generate fixed-length output, called message digests or message-integrity checks (MICs), which act like digital fingerprints on input data. SHA-1 produces a 160-bit hash – its length and structure render current hacking attempts fruitless for reverse engineering to find a given payload. The sender then takes the digest value and sender's private key as input and runs them through the PKCS encryption algorithm to create the digital signature.

When the recipient trading partner receives an EDI-INT message with a digital signature, the payload document and digital signature are stored in different body parts within a MIME multipart/signed structure.⁷ The trading partner processes the received payload through SHA-1 to obtain a digest, and then uses the public key to decrypt the digital signature and obtain the originator's digest. If the two digest values are equal, the recipient can be confident that the intended payload was received intact without alteration due to transmission interference or hacking. Given that only the originating user's private key could have encrypted the digest, the recipient is also assured of the originator's authenticity.

To inform the originator that the recipient has received and processed a message, EDI-INT uses a special response message called the message digest notification (MDN),⁸ which contains three key fields in its body for message validation:

- The `Original-Message-ID` identifies the original message's unique message ID and associates it with the MDN response.
- The `Disposition` value indicates whether the message was properly processed or an error occurred, necessitating retransmission of the payload.
- The `Received-Content-MIC` contains the MIC or digest the recipient obtained when it processed the message. The originator compares this against the digest value computed before transmission to determine whether the payload was safely delivered.

To ensure the MDN's authenticity, the trading partner will apply a digital signature to the MDN itself.

When a returned MDN has a `Disposition` value of `processed`, an expected MIC value, and a digital signature, the originating trading partner can claim *nonrepudiation of receipt*. NRR is a legal event that ensures that the recipient trading part-

ner can't successfully deny having received the message – a beneficial mechanism in preventing disagreements over business transactions.

Transaction Confidentiality

In addition to authentication, content integrity, and message validation, EDI-INT uses S/MIME encryption to provide trading partners with the security of transaction confidentiality. First, the entire message body, including the digital signature, runs through a symmetric encryption algorithm (generally, the Triple Data-Encryption Standard [3DES]) along with a randomly generated session key as the second input (see Figure 1b). Because symmetric algorithms employ just one key for both encryption and decryption, distributing the key among multiple trading partners presents a significant security liability. Instead, a new, random session key is generated each time a trading partner sends an encrypted message. Only the sender and receiver of the EDI-INT message know this key, which isn't used again after the message is sent. The 3DES algorithm provides strong enough cryptography to prevent hacking as long as only the sender and receiver know the session key. Symmetric encryption is also significantly quicker than asymmetric: if EDI-INT required the message body – potentially several megabytes in size – to run through an asymmetric algorithm, the encryption time could take hours rather than seconds.

To work around these dilemmas, EDI-INT uses the session key as an input to the PKCS encryption algorithm along with the public key from the intended recipient's certificate. This produces an encrypted session key that only the receiving trading partner can decrypt. Next, the sender's EDI-INT software encapsulates the encrypted payload and the encrypted session key inside an S/MIME encryption envelope structure to form the encrypted EDI-INT message. Although not necessarily an intuitive approach, encrypting the session key after using it to encrypt the message body provides public-key cryptography's benefits with an acceptable time delay. Because only the recipient's private key can unlock the session key, which will only then unlock the payload, the originator can be confident that only the intended trading partner can view the business transaction.

Standards within EDI-INT

With a robust method for data security and message-receipt validation in place, the EDI-INT

WG began defining standards for Internet transportation. To begin, members drafted applicability statements for each of the Internet's major transports:

- AS1 for email.⁹
- AS2 for Web or HTTP transport,¹⁰ and
- AS3 for FTP.¹¹

Each standard constructs EDI-INT messages with headers that contain the necessary routing and processing information for the underlying transport, as well as instructions for MDN request and construction. Message bodies are structured according to MIME rules; they store the EDI or XML payload and applied digital signature. If S/MIME encryption occurs, the body is encrypted, but the headers remain visible for processing. (Each of the EDI-INT standards includes example messages.)

By borrowing heavily from existing IETF proposed standards, the EDI-INT WG quickly produced viable applicability statements for implementation. In response to feedback from implementers, it added or clarified sections within the drafts to improve understanding and performance and provide explanation regarding how to apply MIME, S/MIME, and MDN.

The WG also made adjustments to each standard according to its Internet transport. For example, AS1 uses email's `To` and `From` headers to identify the originating and receiving trading partners. Because HTTP doesn't use these headers, the AS2 standard adds mandatory `AS2-To` and `AS2-From` headers to its messages to compensate.

AS2 Requirements

Of the EDI-INT standards, AS2 is the most commonly used in B2B transactions. It also contains the most unique elements among the three standards.

As an alternative to S/MIME encryption, AS2 implementers can use the Transport-Layer Security (TLS) protocol over HTTP or with HTTPS, which many companies prefer due its familiarity (HTTPS was a preferred choice for B2B transactions prior to the emergence of AS2 software). HTTPS makes payload encryption redundant because it encrypts the entire transaction, rather than just the message body.

Running over HTTP makes AS2 a true peer-to-peer communication protocol: AS2 applications contain client and server functionality and connect directly to each other. This can provide excellent return time for MDNs. To capitalize on this

fact, the default setting in AS2 is to have message recipients return the MDN synchronously. In such a transmission, the connecting client (originator) keeps the HTTP connection open after it transfers the message to the server (recipient) so that the recipient can parse the message and return the MDN within the HTTP response. This is similar to how a Web browser receives content from a Web site. As a result, the originator generally receives the MDN confirming the transfer within seconds. Yet, the receiving AS2 application might take several minutes or even hours to decrypt and verify the MDN for very large payloads. Given that such delays can cause firewall problems for trading partners, AS2 also lets originators request asynchronous MDN return. In this scenario, the recipient returns only a short HTTP response, without the MDN in the body, before closing the connection. Afterwards, the recipient parses the message, creates the MDN, and then establishes a connection to the originator to return the MDN. The asynchronous MDN arrives as an HTTP request message to which the originating trading partner must send a short HTTP response. The asynchronous MDN looks similar to a normal AS2 message, except that the MDN is stored in its body. To prevent an endless loop, asynchronous MDN messages can't request MDNs for themselves.

Related Standards

Although the EDI-INT working group has focused primarily on the applicability statements, it has also developed a standard to handle compression. Because EDI payloads are often very large, payload compression is important for EDI-INT's growth. Compression can be very useful with HTTPS because TLS adds transaction overhead. Terry Harding created an Internet draft¹² in 2002 to describe how to construct and employ compressed data wrappers to encapsulate EDI documents, using the free and publicly available ZLIB algorithm (www.zlib.net) for the actual data compression. Many EDI-INT applications began making compression part of their default outbound transport settings.

With AS2's growth, the need has also arisen for a means of updating and exchanging digital certificates. After a certificate's listed validity date expires — generally, two to five years after its creation — trading partners should no longer consider it trustworthy. Before a certificate expires, the partners must work out a way to exchange new certificates. A company can easily coordinate and manage this

process out-of-band – via email, for example – when working with only a few trading partners. A company with hundreds of trading partners, on the other hand, needs a more efficient mechanism. Dale Moberg and I recently introduced a certificate-exchange messaging draft¹³ to help solve this dilemma. Although this draft didn't come directly through the EDI-INT WG, group members and EDI-INT standards implementers are fully involved in the ongoing discussion and development.

EDI-INT Acceptance

One of the working group's goals was to facilitate transport interoperability. Even with open standards, however, interoperability problems can result from coding mistakes or different interpretations of standards. To ensure that implementations work together, third-party interoperability tests – executed by Drummond Group and the Uniform Code Council (UCC; www.uc-council.org) – have been present for EDI-INT standards since their inception. While external to the working group's charter, interoperability testing has been a key to its success by independently demonstrating uniformity and acceptance of the standards.

Drummond Group creates interoperability test cases to cover the standard's full scope. For example, one test case requires an AS2 application to send a signed-only message requesting a signed MDN, and another requires the application to send an encrypted and compressed message over HTTPS requesting an asynchronous MDN. Participants must prove that they can send and receive each test case in the test suite with all the other participating implementations. Full-matrix coverage of all test scenarios ensures that each participating product is interoperable with the other EDI-INT implementations tested – some tests included as many as 35 implementations.

AS2 illustrates interoperability testing's value. Several large retailers in the US have now requested that their suppliers use AS2 for all EDI transactions – in large part because successful interoperability tests prove that it actually works. Implementers also benefit from interoperability testing, which offers them a forum for improving products as well as a tool for marketing them.

Throughout the years, countless protocols and standards have been developed and then widely ignored, despite some brilliant technical designs and elegant solutions. They often failed not

because of their content, but because they didn't address real needs in simple, affordable ways. By understanding both the current business environment and the existing solutions' limitations, the EDI-INT working group has provided a useful alternative for EDI transportation.

AS2's adoption in the retail and consumer package-goods industries is now widespread in the US with numerous supply-chain companies, including Wal-Mart and Target, using it with their hundreds of trading partners. By reducing operating costs for electronic trading, AS2 also empowers smaller companies that couldn't afford VANs to participate in EDI transactions. In addition, AS2 is finding a role in recent B2B efforts regarding data synchronization. For example, UCCnet (www.uccnet.org) is using it for trading-partner synchronization of data pools, and the Global Data Synchronization Network (GDSN; www.ean-int.org) consortium of data pools requires its use in their communications. Although still used primarily in North America, AS2 implementations are starting to spread to Europe, South America, and the rest of the world, as well.

Given AS2's success to date, other standards will likely begin leveraging EDI-INT, although such developments probably won't come through the WG. With AS1 now an RFC, the AS2 draft in last review for RFC status, and the AS3 draft soon going to last call, the EDI-INT WG has nearly met its milestones; when it does, the IETF will dissolve it. As in most mature working groups, discussion on the email forum has dwindled to a few emails a month, but the EDI-INT standards' future remains bright. □

Acknowledgments

The EDI-INT working group's success wouldn't have been possible without the efforts of its chair, Rik Drummond, and several other individuals, including Terry Harding, Dale Moberg, Mats Jansson, Carl Hage, Jun Ding, and Karen Rosenthal.

References

1. D. Crocker, *MIME Encapsulation of EDI Objects*, RFC 1767, IETF Internet proposed standard, Mar. 1995; www.ietf.org/rfc/rfc1767.txt.
2. N. Borenstein and N. Freed, *Multipurpose Internet Mail Extensions (MIME), Part One: Format of Internet Message Bodies*, RFC 2045, IETF Internet proposed standard, Dec. 1996; www.ietf.org/rfc/rfc2045.
3. M. Murata, S. St. Laurent, and D. Kohn, *XML Media Types*, RFC 3023, IETF Internet proposed standard, Jan. 2001; www.ietf.org/rfc/rfc3023.txt.
4. B. Ramsdell, *S/MIME Version 3.1 Message Specification*,

- RFC 3851, IETF Internet proposed standard, July 2004; www.ietf.org/rfc/rfc3851.txt.
5. B. Kaliski, *PKCS #7: Cryptographic Message Syntax Version 1.5*, RFC 2315, IETF Internet proposed standard, Mar. 1998; www.ietf.org/rfc/rfc2315.txt.
 6. R. Housley et al., *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, RFC 3280, IETF Internet proposed standard, Apr. 2002, www.ietf.org/rfc/rfc3280.txt.
 7. J. Galvin et al., *Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted*, RFC 1847, IETF Internet proposed standard, Oct. 1995; www.ietf.org/rfc/rfc1847.
 8. T. Hansen and G. Vaudreuil, *Message Disposition Notification*, RFC 3798, IETF Internet proposed standard, May 2004; www.ietf.org/rfc/rfc3798.
 9. T. Harding, R. Drummond, and C. Shih, *MIME-Based Secure Peer-to-Peer Business Data Interchange over the Internet*, RFC 3335, IETF Internet proposed standard, Sept. 2002; www.ietf.org/rfc/rfc3335.txt.
 10. D. Moberg and R. Drummond, "MIME-Based Secure Peer-to-Peer Business Data Interchange Using HTTP," IETF Internet draft, Feb. 2005; work in progress.
 11. T. Harding and R. Scott, "FTP Transport for Secure Peer-to-Peer Business Data Interchange over the Internet," IETF Internet draft, Feb. 2005; work in progress.
 12. T. Harding, "Compressed Data for EDIINT," IETF Internet draft, Feb. 2005; work in progress.
 13. K. Meadors and D. Moberg, "Certificate Exchange Messaging for EDI-INT," IETF Internet draft, Feb. 2005; work in progress.

Kyle Meadors is a principal of test processes at Drummond Group (www.drummondgroup.com). He is responsible for overseeing and administering interoperability testing involving the AS1, AS2, and AS3 standards as well as the Global Data Synchronization Network (GDSN). Meadors received a BS in electrical engineering from Mississippi State University. Contact him at kyle@drummondgroup.com.

PURPOSE The IEEE Computer Society is the world's largest association of computing professionals, and is the leading provider of technical information in the field.

MEMBERSHIP Members receive the monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.

COMPUTER SOCIETY WEB SITE The IEEE Computer Society's Web site, at www.computer.org, offers information and samples from the society's publications and conferences, as well as a broad range of information about technical committees, standards, student activities, and more.

BOARD OF GOVERNORS

Term Expiring 2005: Oscar N. Garcia, Mark A. Grant, Michel Israel, Robit Kapur, Stephen B. Seidman, Kathleen M. Swigger, Makoto Takizawa

Term Expiring 2006: Mark Christensen, Alan Clements, Annie Combelles, Ann Q. Gates, James D. Isaak, Susan A. Mengel, Bill N. Schilit

Term Expiring 2007: Jean M. Bacon, George V. Cybenko, Richard A. Kemmerer, Susan K. (Kathy) Land, Itaru Mimura, Brian M. O'Connell, Christina M. Schober

Next Board Meeting: 10 June 2005, Long Beach, CA

IEEE OFFICERS

President and CEO: W. CLEON ANDERSON

President-Elect: MICHAEL R. LIGHTNER

Past President: ARTHUR W. WINSTON

Executive Director: TBD

Secretary: MOHAMED EL-HAWARY

Treasurer: JOSEPH V. LILLIE

VP, Educational Activities: MOSHE KAM

VP, Pub. Services & Products: LEAH H. JAMIESON

VP, Regional Activities: MARC T. APTER

VP, Standards Association: JAMES T. CARLO

VP, Technical Activities: RALPH W. WYNDRUM JR.

IEEE Division V Director: GENE F. HOFFNAGLE

IEEE Division VIII Director: STEPHEN L. DIAMOND

President, IEEE-USA: GERARD A. ALPHONSE



COMPUTER SOCIETY OFFICES

Headquarters Office

1730 Massachusetts Ave. NW

Washington, DC 20036-1992

Phone: +1 202 371 0101

Fax: +1 202 728 9614

E-mail: bq.ofc@computer.org

Publications Office

10662 Los Vaqueros Cir., PO Box 3014

Los Alamitos, CA 90720-1314

Phone: +1 714 821 8380

E-mail: help@computer.org

Membership and Publication Orders:

Phone: +1 800 272 6657

Fax: +1 714 821 4641

E-mail: help@computer.org

Asia/Pacific Office

Watanabe Building

1-4-2 Minami-Aoyama, Minato-ku

Tokyo 107-0062, Japan

Phone: +81 3 3408 3118

Fax: +81 3 3408 3553

E-mail: tokyo.ofc@computer.org



EXECUTIVE COMMITTEE

President:

GERALD L. ENGEL*

Computer Science & Engineering

Univ. of Connecticut, Stamford

1 University Place

Stamford, CT 06901-2315

Phone: +1 203 251 8431

Fax: +1 203 251 8592

g.engel@computer.org

President-Elect: DEBORAH M. COOPER*

Past President: CARL K. CHANG*

VP, Educational Activities: MURALI VARANASI†

VP, Electronic Products and Services:

JAMES W. MOORE (2ND VP)*

VP, Conferences and Tutorials:

YERVANT ZORIAN†

VP, Chapters Activities:

CHRISTINA M. SCHOBER*

VP, Publications: MICHAEL R. WILLIAMS (1ST VP)*

VP, Standards Activities: SUSAN K. (KATHY) LAND*

VP, Technical Activities: STEPHANIE M. WHITE†

Secretary: STEPHEN B. SEIDMAN*

Treasurer: RANGACHAR KASTURI†

2004-2005 IEEE Division V Director:

GENE F. HOFFNAGLE†

2005-2006 IEEE Division VIII Director:

STEPHEN L. DIAMOND†

2005 IEEE Division V Director-Elect:

OSCAR N. GARCIA*

Computer Editor in Chief: DORIS L. CARVER†

Executive Director: DAVID W. HENNAGE†

* voting member of the Board of Governors

† nonvoting member of the Board of Governors

EXECUTIVE STAFF

Executive Director: DAVID W. HENNAGE

Assoc. Executive Director: ANNE MARIE KELLY

Publisher: ANGELA BURGESS

Assistant Publisher: DICK PRICE

Director, Administration: VIOLET S. DOAN

Director, Information Technology & Services:

ROBERT CARE

Director, Business & Product Development:

PETER TURNER