

TCVN ISO/TS 15000-2 : 2007
ISO/TS 15000-2 : 2004
Xuất bản lần 1

NGÔN NGỮ ĐÁNH DẤU MỞ RỘNG KINH DOANH ĐIỆN TỬ (ebXML)
PHẦN 2: QUY ĐỊNH KỸ THUẬT VỀ DỊCH VỤ THÔNG ĐIỆP (ebMS)

Electronic business eXtensible Markup Language (ebXML)
Part 2: Message service specification (ebMS)

Mục lục	Trang
Tình trạng tiêu chuẩn.....	6
Khái quát nội dung tiêu chuẩn.....	7
1 Phạm vi áp dụng.....	11
1.1 Sự phù hợp.....	11
1.2 Tài liệu liên quan.....	12
Chương 1: Chức năng chính.....	13
2 ebXML và SOAP.....	13
2.1 Quy định việc đóng gói.....	13
2.2 Ngôn ngữ khai báo Prolog của XML (XML Prolog).....	16
2.3 Các đuôi mở rộng SOAP của ebXML.....	17
3 Phần tử đuôi mở rộng lõi.....	21
3.1 Phần tử MessageHeader (Tiêu đề thông điệp).....	21
3.2 Phần tử Manifest (Bảng kê).....	27
4 Các môđun lõi.....	29
4.1 Môđun an ninh.....	29
4.2 Môđun điều khiển lõi.....	37
4.3 Môđun SyncReply (trả lời đồng bộ).....	42
5 Sự kết nối các phần tử đuôi mở rộng SOAP của ebXML.....	43
Chương II: Các tính năng bổ sung.....	44
6 Môđun thông điệp xác thực.....	44
6.1 Kho lưu trữ thường xuyên và hệ thống tương thích (Bộ lưu trữ lâu dài và lõi hệ thống).....	44
6.2 Các phương pháp thực hiện đối với thông điệp xác thực.....	45
6.3 Các đuôi mở rộng Header (Tiêu đề) SOAP của thông điệp tin cậy.....	45
6.4 Các tham số truyền thông điệp tin cậy.....	48
6.5 Giao thức truyền thông điệp tin cậy trong ebXML.....	50
6.6 Các kết hợp của việc truyền thông điệp tin cậy.....	56
7 Dịch vụ báo cáo trạng thái thông điệp.....	56
7.1 Các thông điệp báo cáo trạng thái thông điệp.....	57
7.2 Phần tử StatusRequest (yêu cầu trạng thái).....	58
7.3 Phần tử StatusResponse (phản hồi trạng thái).....	59
8 Dịch vụ ping của trình quản lý dịch vụ thông điệp.....	60
8.1 Thông điệp ping của trình quản lý dịch vụ.....	61
8.2 Thông điệp Pong của trình quản lý dịch vụ thông điệp.....	62
8.3 Các xem xét về an ninh.....	63
9 Môđun MessageOrder (Thứ tự thông điệp).....	63

TCVN ISO/TS 15000-2 : 2007

9.1	Phần tử MessageOrder (thứ tự thông điệp).....	63
10	Môđun Multi-Hop (đa bước truyền).....	65
10.1	Việc truyền thông điệp tin cậy Multi-hop (đa bước truyền)	65
10.2	Sắp xếp thứ tự thông điệp và Multi-Hop (đa bước truyền)	68
	Chương III: Các phụ lục chuẩn	69
	Phụ lục A.....	69
	Phụ lục B.....	74
B.1	Giới thiệu	74
B.2	HTTP	74
B.3	SMTP	78
B.4	Lỗi truyền thông trong thông điệp xác thực	82
	Phụ lục C.....	83
	Tài liệu tham khảo	85
	(quy định).....	85
	Tài liệu tham khảo	88
	(tham khảo).....	88

Lời nói đầu

TCVN ISO/TS 15000-2 : 2007 hoàn toàn tương đương với tiêu chuẩn **ISO/TS 15000-2 : 2004**.

TCVN ISO/TS 15000-2 : 2007 do Ban kỹ thuật tiêu chuẩn TCVN/TC 154 "*Quá trình, các yếu tố dữ liệu và tài liệu trong thương mại, công nghiệp và hành chính*" biên soạn, Tổng cục Tiêu chuẩn Đo lường Chất lượng trình duyệt, Bộ Khoa học và Công nghệ công bố.

Tình trạng tiêu chuẩn

Tiêu chuẩn này quy định thông điệp ebXML đối với cộng đồng kinh doanh điện tử (eBusiness). Tiêu chuẩn này dựa trên cơ sở dạng RFC tiêu chuẩn của Internet Society (cộng đồng người sử dụng Internet) được chuyển sang dạng Microsoft Word 2000.

CHÚ THÍCH Người thực thi tiêu chuẩn này nên tham khảo các soát xét tiêu chuẩn, tình trạng của tiêu chuẩn trên trang web của Ban kỹ thuật về dịch vụ thông điệp ebXML của OASIS (<http://www.oasis-open.org/committees/ebxml-msg/>).

Bên tham gia xây dựng quy định kỹ thuật về dịch vụ thông điệp ebXML của OASIS

Ralph Berwanger Individual Member

Dick Brooks Individual Member

Doug Bunting Sun Microsystems, Inc

David Burdett Commerce One

Arvola Chan TIBCO

Sanjay Cherian Sterling Commerce

Cliff Collins Sybase

Philippe DeSmedt Individual Member

Colleen Evans Sonic Software

Chris Ferris Sun Microsystems, Inc

David Fischer Drummond Group

Jim Galvin Drummond Group

Brian Gibb Sterling Commerce

Scott Hinkelman IBM

Jim Hughes Hewlett Packard

Kazunori Iwasa Fujitsu Limited

Ian Jones Individual Member

Brad Lund Intel™ Corporation

Bob Miller GE Global eXchange

Dale Moberg Cyclone Commerce

Himagiri Mukkamala Sybase

Bruce Pedretti Hewlett-Packard

Yukinori Saito Individual Member

Martin Sachs IBM Research

Jeff Turpin Cyclone Commerce

Aynur Unal E2Open

Cedrec Vessell DISA

Daniel Weinreb eXcelon

Pete Wenzel SeeBeyond

Prasad Yendluri WebMethods

Sinisa Zimek SAP.

Khái quát nội dung tiêu chuẩn

Tiêu chuẩn này định nghĩa giao thức dịch vụ thông điệp ebXML cho phép trao đổi thông điệp giữa hai bên tham gia một cách an toàn và tin cậy, bao gồm các mô tả sau:

- cấu trúc thông điệp ebXML được sử dụng để gói dữ liệu vùng mang thông tin truyền tải giữa hai bên tham gia;
- cách gửi và nhận các thông điệp của trình quản lý dịch vụ thông điệp qua một giao thức truyền thông dữ liệu.

Tiêu chuẩn này độc lập với giao thức truyền thông và vùng mang thông tin được sử dụng. Các phụ lục trong tiêu chuẩn này mô tả cách sử dụng tiêu chuẩn này cùng với HTTP [RFC2616] và SMTP [RFC2821].

Tiêu chuẩn này bao gồm các mục sau:

Chức năng chính

- **Quy định về đóng gói** (Packaging Specification) - Mô tả cách đóng gói một thông điệp ebXML và các phần tương ứng của nó trong một biểu mẫu có thể được gửi nhờ việc sử dụng một giao thức truyền thông như HTTP hoặc SMTP;
- **Các đuôi mở rộng của đường bao SOAP ebXML** (ebXML SOAP Envelop) - Quy định về cấu trúc và các phần tử hỗn hợp của thông tin cần thiết cho một dịch vụ thông điệp ebXML để tạo hoặc xử lý một thông điệp ebXML;
- **Quản lý lỗi** - Mô tả cách một dịch vụ thông điệp ebXML báo cáo các lỗi được phát hiện cho trình quản lý dịch vụ thông điệp ebXML khác (phần 4.2);
- **An ninh** - Cung cấp một quy định các ngữ nghĩa an ninh cho các thông điệp ebXML (phần 4.1);

TCVN ISO/TS 15000-2 : 2007

- **SyncReply** (trả lời đồng bộ) - Chỉ ra MSH tiếp theo (Next MSH) trả lời lại là đồng bộ hay không đồng bộ.

Các đặc tính bổ sung

- **Truyền thông điệp tin cậy** - Chức năng truyền thông điệp tin cậy xác định một giao thức khả năng hoạt động tương tác tại bất kỳ hai phần mềm thực thi dịch vụ thông điệp có thể trao đổi các thông điệp được gửi có sử dụng các ngữ nghĩa truyền once-and-only-once (đồng thời và chỉ đồng thời) (phần 6);
- **Dịch vụ báo cáo tình trạng thông điệp** - Mô tả các dịch vụ cho phép một dịch vụ phát hiện ra tình trạng của trình quản lý thông điệp khác (MSH) hoặc một thông điệp cá nhân;
- **Trình tự thông điệp** - Thứ tự của thông điệp nhận bởi To Party MSH (MSH bên tham gia To) có thể được đảm bảo;
- **Multi-Hop** (Đa bước truyền) - Các thông điệp có thể được gửi thông qua các nút MSH trung gian (phần 10).

Các phụ lục trong tiêu chuẩn này:

- **phụ lục A Giản đồ** - Đây là phụ lục quy định; bao gồm định nghĩa giản đồ XML (XMLSchema) cho các đuôi mở rộng của **Header** (tiêu đề) và **Body** (thân) của SOAP ebXML;
- **phụ lục B Các phép ánh xạ đường bao giao thức truyền thông** - Đây là phụ lục quy định; mô tả cách truyền các thông điệp phù hợp với dịch vụ thông điệp ebXML qua HTTP và SMTP;
- **phụ lục C Hồ sơ an ninh** - Đề cập về các hồ sơ dịch vụ an ninh.

Quy ước trong tiêu chuẩn

Các thuật ngữ in nghiêng được định nghĩa trong bảng chú giải thuật ngữ [ebGLOSS]. Các thuật ngữ nghiêng đậm trình bày nội dung thuộc tính và/hoặc phần tử. Thuật ngữ được liệt kê dưới dạng phong chữ Courier để cập đến các thành phần MIME. Các chú thích được liệt kê dưới dạng phong chữ Times New Roman là để tham khảo. Tên các thuộc tính được bắt đầu bằng chữ thường. Tên các phần tử được bắt đầu bằng chữ hoa.

Các từ khóa MUST (phải), MUST NOT (không phải), REQUIRED (yêu cầu), SHALL (phải), SHALL NOT (không phải), SHOULD (nên), SHOULD NOT (không nên), RECOMMENDED (khuyến cáo), MAY (có thể) và OPTION (tùy chọn), khi xuất hiện trong tiêu chuẩn này được dịch như đã quy định trong [RFC2119] như được nêu ra ở đây:

- MUST, SHALL (PHẢI) hoặc REQUIRED (YÊU CẦU) có nghĩa là định nghĩa đó là một yêu cầu tuyệt đối trong tiêu chuẩn này;
- MUST NOT, SHALL NOT (KHÔNG PHẢI) định nghĩa đó là một sự ngăn cấm trong tiêu chuẩn này;

- SHOULD (NÊN), RECOMMENDED (KHUYẾN CÁO) có nghĩa có một số lý do hợp lệ trong các trường hợp cụ thể để bỏ qua các mục cụ thể, nhưng các hàm ý đầy đủ phải được hiểu và đánh giá cẩn thận trước khi chọn một tiến trình khác;
- SHOULD NOT (KHÔNG NÊN), NOT RECOMMENDED (KHÔNG KHUYẾN CÁO) có nghĩa là có thể tồn tại một số lý do hợp lệ trong các trường hợp cụ thể khi hành vi cụ thể được chấp nhận hay hữu ích, nhưng các hàm ý đầy đủ nên được hiểu và đánh giá trong các trường hợp cụ thể trước khi thực thi bất kỳ hành vi được mô tả nào đó trong trường hợp này;
- MAY (CÓ THỂ), OPTION (TÙY CHỌN) có nghĩa một mục là tùy chọn đúng. Một nhà cung cấp có thể chọn lựa để bao gồm mục này bởi vì một thị trường cụ thể yêu cầu nó hoặc do nhà cung cấp đó cảm thấy nó tăng cường sản phẩm trong khi một nhà cung cấp khác có thể loại bỏ mục đó.

Một phần mềm thực thi không bao gồm mục tùy chọn được chuẩn bị để có khả năng hoạt động tương tác với các phần mềm thực thi khác có bao gồm tùy chọn đó, mặc dù có thể cùng với chức năng bị cắt giảm. Trong cùng đặc điểm một phần mềm thực thi không bao gồm tùy chọn (loại bỏ, một vấn đề dĩ nhiên, đối với đặc tính mà tùy chọn đó cung cấp).

Ngôn ngữ đánh dấu mở rộng trong kinh doanh điện tử (ebXML)

Phần 2: Quy định thoả thuận và hồ sơ thủ tục hợp tác (ebCPP)

Electronic business eXtensible Markup Language (ebXML)

Part 2: Message service specification (ebMS)

1 Phạm vi áp dụng

Dịch vụ thông điệp ebXML (ebMS) xác định giản đồ tài liệu tiêu đề và đường bao thông điệp được sử dụng để truyền thông điệp ebXML bằng một giao thức truyền thông như HTTP (Hypertext Transfer Protocol - Giao thức truyền siêu văn bản) hoặc SMTP (Simple Mail Transfer Protocol - Giao thức truyền thư điện tử đơn giản) và cách hoạt động của phần mềm gửi và nhận thông điệp ebXML. ebMS là một tập các đuôi mở rộng phân tầng cho giao thức truy cập đối tượng cơ bản [SOAP - Simple Object Access Protocol] và thông điệp SOAP cùng với các quy định kèm theo [SAOPAttach]. Tiêu chuẩn này cung cấp các tính năng tin cậy và an ninh cần thiết cho thương mại điện tử quốc tế. Các tính năng tin cậy và an ninh này không được cung cấp trong SOAP hoặc SOAPAttach.

Cơ sở hạ tầng ebXML gồm nhiều thành phần độc lập có liên quan. Các quy định về các thành phần riêng này được trình bày trong các tài liệu độc lập. Các quy định đó tự nó đã bao gồm các thành phần, tuy nhiên, các quyết định thiết kế trong một tài liệu có thể và thực sự ảnh hưởng đến tài liệu khác. Xét trên khía cạnh này, ebMS là một định nghĩa kết hợp chặt chẽ đối với một trình quản lý dịch vụ thông điệp (MSH - Trình quản lý dịch vụ thông điệp).

ebMS cung cấp các phương tiện truyền, định tuyến và đóng gói thông điệp (Message Package) đối với hạ tầng ebXML. ebMS không phải là một thành phần vật lý mà là một khái niệm trừu tượng của quá trình. Việc thực thi theo tiêu chuẩn này hoàn toàn là sự ứng dụng phần mềm độc lập hoặc một thành phần tích hợp của một số quá trình thương mại rộng hơn.

1.1 Sự phù hợp

Sự thực thi phù hợp với tiêu chuẩn PHẢI thỏa mãn tất cả các điều kiện sau:

- hỗ trợ mọi cú pháp, tính năng và cách hoạt động bắt buộc (xác định bằng các từ khóa [RFC2119] PHẢI, KHÔNG ĐƯỢC, YÊU CẦU) trong Phần I – Chức năng chính;
- hỗ trợ mọi cú pháp, tính năng và cách hoạt động bắt buộc trong Phần II – Tính năng bổ sung;
- tuân theo việc thực thi sau đây của các từ khóa TỰ CHỌN và CÓ THỂ: Khi các từ khóa này áp dụng cho cách hoạt động của thực thi, thực thi có thể hỗ trợ các cách hoạt động này hoặc không,

xem [RFC2119]. Khi các từ TỰ CHỌN và CÓ THỂ áp dụng cho nội dung thông điệp liên quan đến một môđun đặc tính thì sự phù hợp với thực thi các đặc tính như vậy PHẢI có khả năng xử lý nội dung thông điệp tùy chọn này theo ngữ nghĩa ebXML được mô tả;

- việc thực thi cú pháp, đặc tính và/hoặc cách hoạt động tùy chọn được xác định trong tiêu chuẩn này PHẢI có khả năng hoạt động tương tác với thực thi khác chưa thực thi cú pháp, đặc tính và/hoặc cách hoạt động tùy chọn đó. Nó PHẢI có khả năng xử lý cơ chế lỗi quy định đối với các đặc tính tùy chọn được chọn để thực thi;
- có khả năng hoạt động tương tác với thực thi khác đã được chọn để thực thi cú pháp, đặc tính và/hoặc hoạt động tùy chọn được xác định trong tiêu chuẩn này. Việc thực hiện các đặc tính không hỗ trợ PHẢI theo cơ chế quy định đối với đặc tính đó.

1.2 Tài liệu liên quan

ebXML Technical Architecture Specification (Quy định kiến trúc kỹ thuật ebXML) [ebTA] - Xác định kiến trúc kỹ thuật tổng thể đối với ebXML;

ebXML Technical Architecture Risk Assessment Technical Report (Báo cáo kỹ thuật về đánh giá rủi ro của kiến trúc kỹ thuật ebXML) [secRISK] - Xác định các cơ chế an ninh cần thiết để phủ nhận các mối đe dọa được lựa chọn và dự trước;

ebXML Collaboration Protocol Profile và Agreement Specification (Quy định về thỏa thuận và hồ sơ thủ tục hợp tác ebXML) [ebCPP] - Xác định cách thức một bên tham gia có thể phát hiện và/hoặc thỏa thuận thông tin mà bên tham gia đó cần để hiểu bên tham gia khác trước khi gửi cho bên tham gia khác thông điệp tuân theo tiêu chuẩn này;

ebXML Registry/Repository Services Specification (Quy định các dịch vụ về kho/sổ đăng ký ebXML) [ebRS] - Xác định một dịch vụ đăng ký đối với môi trường ebXML.

Chương 1: Chức năng chính

2 ebXML và SOAP

Quy định dịch vụ thông điệp ebXML là một tập hợp các đuôi mở rộng của phần tử **Header** (tiêu đề) và **Body** (thân) của SOAP có tên miền được hạn định trong **Envelope** (đường bao) của SOAP. Chúng được đóng gói trong một MIME (Multipurpose Internet Mail Extension - Thư Internet toàn năng mở rộng) đa phần để cho phép chứa các vùng mang thông tin (payload) hoặc thông tin đính kèm (attachments) với các phần tử đuôi mở rộng SOAP. Nói chung, các phần tử đuôi mở rộng SOAP của ebXML được sử dụng khi:

- sử dụng các phần mềm khác nhau để tạo ra các phần tử đuôi mở rộng SOAP của ebXML;
- Một phần tử đuôi mở rộng SOAP của ebXML không tồn tại hoặc;
- dữ liệu trong phần tử đuôi mở rộng SOAP của ebXML được ký số phân biệt với các phần tử đuôi mở rộng SOAP của ebXML khác.

2.1 Quy định việc đóng gói

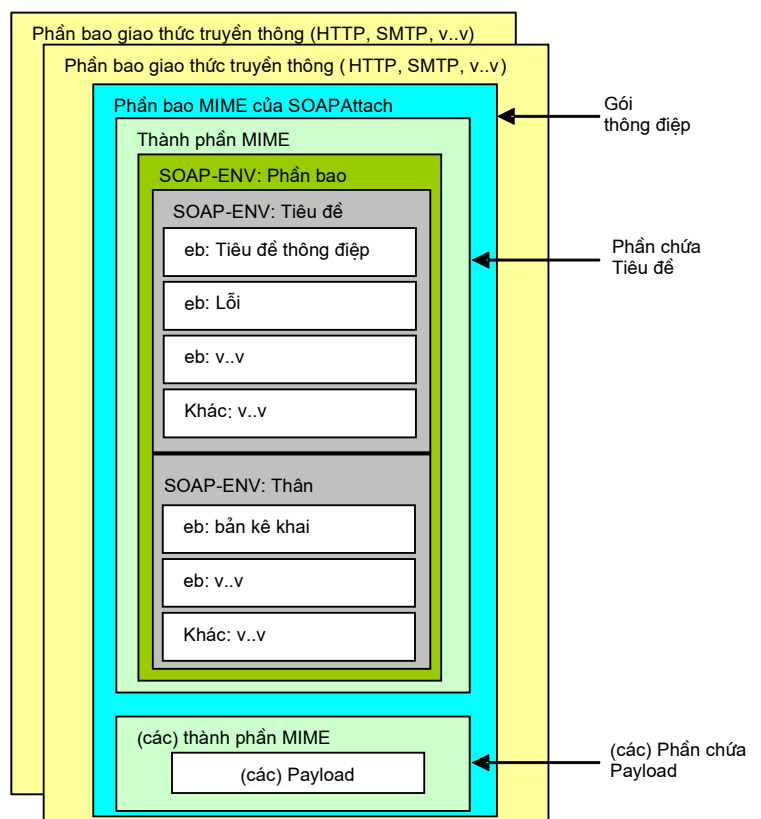
Một thông điệp ebXML là một giao thức truyền thông độc lập với đường bao thông điệp chung đa phần/MIME (Multipurpose Internet Mail Extension - Thư Internet toàn năng mở rộng) có cấu trúc theo thông điệp SOAP có đính kèm [SOAP Attach] hoặc *gói thông điệp (Message Package)*.

Có hai phần MIME logic trong *gói thông điệp (Message Package)*:

- Phần thứ nhất là phần chức tiêu đề bao gồm một thông điệp phù hợp SOAP 1.1. Trong phần còn lại của tiêu chuẩn này, tài liệu XML được coi là một thông điệp SOAP.
- Phần thứ hai là không hoặc nhiều thành phần MIME bổ sung gọi là các *phần chứa vùng mang thông tin*, bao gồm các vùng mang thông tin về mức ứng dụng.

Cấu trúc tổng quát và hỗn hợp của một thông điệp ebXML được mô tả trong hình (2.1).

Thông điệp SOAP là một tài liệu XML bao gồm Một phần tử **Envelope** SOAP. Phần tử gốc của tài liệu XML này biểu diễn một thông điệp SOAP. Phần tử **Envelope** SOAP (đường bao SOAP) bao gồm:



Hình 2.1 - Cấu trúc thông điệp ebXML

TCVN ISO/TS 15000-2 : 2007

- Một phần tử **Header** SOAP, đây là cơ chế chung để bổ sung các đặc tính cho một thông điệp SOAP, bao gồm các phần tử tiêu đề ebXML cụ thể.
- Một phần tử **Body** SOAP, đây là một vùng dành cho trình điều khiển dịch vụ thông điệp điều khiển dữ liệu và thông tin liên quan đến các phần chứa vùng mang thông tin (phần chứa vùng mang thông tin (Payload Container)) của thông điệp.

2.1.1 Sự phù hợp cấu trúc SOAP

Thông điệp ebXML đóng gói theo:

- giao thức truy cập đối tượng đơn giản 1.1 (Simple Object Access Protocol - SOAP) [SOAP];
- các thông điệp SOAP có đính kèm [SOAPAttach].

Việc mang các tiêu đề ebXML trong *thông điệp* SOAP có nghĩa là ngữ nghĩa SOAP của ebXML ánh xạ trực tiếp lên ngữ nghĩa SOAP.

2.1.2 Gói thông điệp

Tất cả các phần tử tiêu đề MIME của *gói thông điệp* (*Message Package*) tuân theo quy định thông điệp SOAP có đính kèm [SOAP Attach]. Hơn nữa, tiêu đề MIME `Content-type` trong *gói thông điệp* (*Message Package*) bao gồm một thuộc tính `type` (kiểu) phù hợp kiểu môi trường truyền MIME của phần thân MIME bao gồm tài liệu *thông điệp* SOAP. Theo quy định [SOAP] thì kiểu MIME của *thông điệp* SOAP có giá trị "text/xml".

KHUYẾN CÁO là các tiêu đề khởi đầu bao gồm một tiêu đề MIME `Content-ID` có cấu trúc theo MIME [RFC2045], tham số `start` (TUỖ CHỌN trong Multipart/Related (Đa phần/Liên quan) của MIME [RFC2387]) luôn tồn tại trong các tham số yêu cầu đối với kiểu phương tiện multipart/related (đa phần/liên quan) giúp thêm việc loại bỏ lỗi tìm thấy. Sau đây là một ví dụ về tiêu đề MIME đối với *gói thông điệp* (*Message Package*) multipart/related:

```
Content-Type: multipart/related; type="text/xml"; boundary="boundaryValue";
start=messagepackage-123@example.com
```

```
--boundaryValue
Content-ID: <messagepackage-123@example.com>
```

Các thực thi PHẢI hỗ trợ các thông điệp không đa phần (non-multipart), xuất hiện khi không có các vùng mang thông tin ebXML. Một thông điệp ebXML không có vùng mang thông tin có thể được gửi như một thông điệp SOAP đơn giản hoặc một thông điệp đa phần [SOAP Attach] có chỉ một phần thân.

2.1.3 Phần chứa tiêu đề

Phần thân gốc của *gói thông điệp* (*Message Package*) là *phần chứa tiêu đề*. *Phần chứa tiêu đề* là một phần thân MIME bao gồm một thông điệp SOAP được quy định trong quy định thông điệp SOAPAttach.

2.1.3.1 Content-Type

Tiêu đề Content-Type của MINE cho *phần chứa tiêu đề (Header Container)* PHẢI có giá trị "text/xml" theo quy định [SOAP]. Tiêu đề Content-Type CÓ THỂ bao gồm một thuộc tính "charset". Ví dụ:

```
Content-Type: text/xml; charset="UTF-8"
```

2.1.3.2 Thuộc tính charset

Thuộc tính charset của MINE đặc trưng cho bộ ký tự được dùng để tạo *thông điệp SOAP*. Ngữ nghĩa của thuộc tính được mô tả trong "các điều kiện Mã hóa /tham số (encoding/parameter) charset" của text/xml trong XML [XMLMedia]. Danh sách các giá trị hợp lệ có thể xem tại <http://www.iana.org/>.

Nếu tồn tại cả hai điều kiện thì thuộc tính charset MINE PHẢI tương đương với khai báo mã hóa *thông điệp SOAP*, thuộc tính charset MINE PHẢI không bao gồm một giá trị xung đột với mã tạo ra *thông điệp SOAP*.

KHUYẾN CÁO sử dụng UTF-8 [UTF-8] để mã hoá tài liệu này cho khả năng hoạt động tương tác tối đa. Thuộc tính MINE này không mặc định trong các quy tắc xử lý dành cho kiểu trung gian từ text/xml [XMLMedia].

2.1.3.3 Ví dụ phần chứa tiêu đề

Đoạn dưới đây là một ví dụ về *phần chứa tiêu đề*.

```
Content-ID: <messagepackage-123@example.com>
Content-Type: text/xml; charset="UTF-8"

<SOAP:Envelope
  xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
  <SOAP:Header>
  ...
</SOAP:Header>
<SOAP:Body>
  ...
</SOAP:Body>
</SOAP:Envelope>
--boundaryValue
```

---| Header

--| SOAP Message

2.1.4 Phần chứa vùng mang thông tin (payload container)

Có không hoặc nhiều *vùng mang thông tin* trong một *gói thông điệp (Message Package)* phù hợp với quy định thông điệp SOAP có đính kèm [SOAPAttach].

Nếu *gói thông điệp* đó có một vùng mang thông tin ứng dụng (application payload) thì nó NÊN được kèm theo trong một *phần chứa vùng mang thông tin* (phần chứa vùng mang thông tin (Payload Container)).

Nếu không có vùng mang thông tin ứng dụng (application payload) trong *gói thông điệp (Message Package)* thì không tồn tại *phần chứa vùng mang thông tin* (phần chứa vùng mang thông tin (Payload Container)).

TCVN ISO/TS 15000-2 : 2007

Nội dung của mỗi *phần chứa vùng mang thông tin* (phần chứa vùng mang thông tin (Payload Container)) PHẢI được định danh trong phần tử **Manifest** của thông điệp ebXML trong phần tử **Body** SOAP (xem mục 3.2).

Quy định dịch vụ thông điệp ebXML không hạn chế về cấu trúc hoặc nội dung các *vùng mang thông tin ứng dụng* (application payload). Vùng mang thông tin có thể là văn bản thông thường hoặc đối tượng đa phần phức tạp. Quy định cấu trúc và vị trí các đối tượng vùng mang thông tin thuộc quyền của cơ quan quy định quá trình thương mại hoặc trao đổi thông tin sử dụng *dịch vụ thông điệp ebXML*.

2.1.4.1 Ví dụ phần chứa vùng mang thông tin (phần chứa vùng mang thông tin (Payload Container))

Đoạn sau trình bày một ví dụ một *phần chứa vùng mang thông tin* (phần chứa vùng mang thông tin (Payload Container)) và một vùng mang thông tin:

Content-ID: <domainname.example.com>	-----	ebXML MIME	
Content-Type: application/xml	-----		
<Invoice>	-----		Payload Container
<Invoicedata>		Payload	
...			
</Invoicedata>			
</Invoice>	-----		

CHÚ THÍCH: Content-type trong ví dụ trước (application/XML) khác content-type trong ví dụ SOAP phần 1.1.2 (text/XML) ở trên. SOAP 1.1 quy định trạng thái content-type sử dụng cho SOAP PHẢI là 'text/XML'. Tuy nhiên, nhiều chuyên gia MIME không đồng ý với kiểu thiết kế 'text/*' đối với tài liệu XML do "con người không đọc được" phần lớn XML trong khi sự thiết kế MIME dạng 'văn bản' ('text') lại có nghĩa. Họ cho rằng tài liệu XML nên có dạng 'application/XML'.

2.1.5 Các tham số MIME bổ sung

Bất kỳ phần MIME nào được quy định ở đây cũng có thể bao gồm thêm các tiêu đề MIME theo quy định MIME [RFC2045]. Việc thực thi có thể lược bỏ bất kỳ tiêu đề MIME nào không được quy định ở đây. Việc thực thi PHẢI bỏ qua bất kỳ tiêu đề MIME không công nhận nào.

Ví dụ một sự thực thi có thể gồm content-length trong một thông điệp. Tuy nhiên bên nhận thông điệp này có thể lược bỏ content-length.

2.1.6 Báo cáo lỗi MIME

Nếu một lỗi MIME được tìm thấy trong *gói thông điệp (Message Package)* thì nó PHẢI được báo cáo theo quy định trong SOAP có đính kèm [SOAPAttach].

2.2 Ngôn ngữ khai báo Prolog của XML (XML Prolog)

Ngôn ngữ khai báo Prolog của XML (XML Prolog) của *thông điệp SOAP* CÓ THỂ bao gồm một khai báo XML. Tiêu chuẩn này không quy định các chú thích bổ sung hoặc chỉ dẫn xử lý trong ngôn ngữ khai báo Prolog của XML (XML Prolog). Ví dụ:

```
Content-Type: text/xml; charset="UTF-8"  
<?xml version="1.0" encoding="UTF-8"?>
```


2.2.1 Khai báo XML

Khai báo XML CÓ THỂ có trong một thông điệp SOAP PHẢI bao gồm đặc tả phiên bản mà các khuyến cáo XML [XML] yêu cầu và CÓ THỂ bao gồm một khai báo việc mã hóa. Một dịch vụ thông điệp ebXML PHẢI được thực thi bởi *dịch vụ thông điệp ebXML* phù hợp.

2.2.2 Khai báo việc mã hóa

Nếu có cả bộ ký tự *phần chứa tiêu đề (Header Container)* và khai báo việc mã hóa MINE thì ngôn ngữ khai báo Prolog của XML (XML Prolog) cho *thông điệp SOAP PHẢI* bao gồm khai báo việc mã hóa tương thích với thuộc tính `charset` của `Content-type MINE` của *phần chứa tiêu đề (Header Container)* (xem phần 2.1.3).

Nếu khai báo việc mã hóa KHÔNG PHẢI bao gồm một giá trị xung đột với mã tạo *thông điệp SOAP*. Thì KHUYẾN CÁO sử dụng UTF-8 để mã hoá *thông điệp SOAP*.

Nếu thuộc tính mã không thể xác định được bởi bộ xử lý XML sử dụng các quy tắc trong phần 2.3.3 của XML [XML] thì khai báo XML và khai báo mã của nó PHẢI được cung cấp trong tài liệu tiêu đề SOAP của ebXML.

CHÚ THÍCH: Khai báo việc mã hóa không được yêu cầu trong một tài liệu XML theo quy định XML v1.0 [XML].

2.3 Các đuôi mở rộng SOAP của ebXML

Theo quy định [SOAP], tất cả nội dung phần tử đuôi mở rộng là tên miền hạn định. Tất cả nội dung phần tử đuôi mở rộng SOAP của ebXML quy định ở đây được hạn định là tên miền đuôi mở rộng **Envelope** của SOAP (đường bao SOAP) của ebXML được quy định trong phần 2.2.2.

Các khai báo tên miền (các thuộc tính `xmlns pseudo`) cho các đuôi mở rộng SOAP của ebXML có thể nằm trong các phần tử **Body**, **Header** hoặc **Envelope** SOAP (đường bao SOAP) của ebXML hoặc trực tiếp trong các phần tử đuôi mở rộng SOAP của ebXML.

2.3.1 Thuộc tính giả tên miền (pseudo)

Khai báo tên miền cho đuôi mở rộng **Envelope** SOAP (đường bao SOAP) của ebXML (thuộc tính giả **xmlns**) (xem [XMLNS]) có giá trị YÊU CẦU sau đây:

`http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd`

2.3.2 Thuộc tính `xsi:schemaLocation` (Vị trí giản đồ)

Tên miền SOAP:

`http://schemas.xmlsoap.org/soap/envelope/`

TCVN ISO/TS 15000-2 : 2007

Liên quan đến một quy định giản đồ XML của W3C. TC truyền thông điệp ebXML của OASIS ebXML cung cấp một phiên bản giản đồ SOAP tương ứng phù hợp với phiên bản khuyến cáo W3C về quy định giản đồ XML [XMLSchema].

Mọi thực thi MSH ebXML được KHUYẾN CÁO áp dụng tên miền XMLSchema-instance với thuộc tính **schemaLocation** trong phần tử **Envelope** SOAP (đường bao SOAP) chỉ tới ra vị trí cú pháp hợp lệ của tài liệu giản đồ và tài liệu. Việc không áp dụng thuộc tính **shemaLocation** hạn chế tính hợp lệ của giản đồ XML trong các thông điệp nhận được

Ví dụ:

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
               http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd">
```

Hơn nữa, nội dung phần tử đuôi mở rộng **Header** và **Body** SOAP của ebXML có thể giúp sự phân tích cú pháp thấy được nội dung tài liệu giản đồ có bao gồm tên miền hạn định các phần tử đuôi mở rộng SOAP.

Giản đồ phần tử đuôi mở rộng SOAP ebXML sử dụng phiên bản khuyến cáo W3C về quy định giản đồ XML [Giản đồ XML] (xem Phụ lục A). Tên miền XML Schema-instance hạn định thuộc tính **schemaLocation** phải gồm một ánh xạ từ tên miền mở rộng **Envelope** SOAP (đường bao SOAP) của ebXML tới tài liệu giản đồ của nó trong và phần tử khai báo không tên miền đuôi mở rộng **Envelope** SOAP (đường bao SOAP) của ebXML.

SchemaLocation (vị trí lược đồ) cho tên miền trong phần 2.3.1 là:

```
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd
```

Thuộc tính schemaLocation tách biệt được KHUYẾN CÁO sử dụng, có thể không như thuộc tính **schemaLocation** đối với giản đồ hơn một tên miền, vẫn hợp lệ với một thông điệp SOAP của ebXML.

Ví dụ:

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
               http://www.oasis-open.org/committees/ebxml-
               msg/schema/envelope.xsd">
  <SOAP:Header
    xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
    header-2_0.xsd"
    xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-
    msg/schema/msg-header-2_0.xsd
    http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
    header-2_0.xsd">
    <eb:MessageHeader ...>
      ...
    </eb:MessageHeader>
  </SOAP:Header>
  <SOAP:Body
    xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
    header-2_0.xsd"
```

```

xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-
msg/schema/msg-header-2_0.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
header-2_0.xsd">
<eb:Manifest eb:version="2.0">
...
</eb:Manifest>
</SOAP:Body>
</SOAP:Envelope>

```

2.3.3 Phần tử Header SOAP

Phần tử **Header** SOAP là phần tử con đầu tiên của phần tử **Envelope** SOAP. Nó **PHẢI** có một hạn định tên miền phù hợp với sự khai báo tên miền **Envelope** SOAP (đường bao SOAP) đối với tên miền "http://schemas.xmlsoap.org/soap/envelope/".

2.3.4 Phần tử Body SOAP

Phần tử **Body** SOAP là phần tử con thứ hai của phần tử **Envelope** SOAP. Nó **PHẢI** có một hạn định tên miền phù hợp sự với khai báo tên miền **Envelope** SOAP (đường bao SOAP) đối với tên miền "http://schemas.xmlsoap.org/soap/envelope/".

2.3.5 Đuôi mở rộng SOAP của ebXML

Một thông điệp ebXML mở rộng *thông điệp* SOAP có các phần tử đuôi mở rộng sau:

2.3.5.1 Đuôi mở rộng Header của SOAP

- **MessageHeader** (tiêu đề thông điệp) – Một phần tử **YÊU CẦU** bao gồm thông tin định tuyến cho thông điệp (đến/từ, v.v.) và các thông tin khác về thông điệp;
- **SyncReply** (trả lời đồng bộ) – Một phần tử chỉ trạng thái truyền tải yêu cầu tới nút SOAP tiếp theo.

2.3.5.2 Đuôi mở rộng Body của SOAP

- **Manifest** – Một phần tử đánh dấu bất kỳ dữ liệu nào có mặt hoặc trong các phần chứa *phần chứa vùng mang thông tin* (phần chứa vùng mang thông tin (Payload Container)) hoặc phần khác, ví dụ trên web. **CÓ THỂ** lược bỏ phần tử này.

2.3.5.3 Môđun lỗi của ebXML

- môđun kiểm soát lỗi (Error Handling Module);
- **ErrorList** (danh sách lỗi) – Một phần tử tiêu đề SOAP bao gồm một danh sách lỗi được báo cáo dựa vào một thông điệp trước đó. Chỉ được sử dụng phần tử **ErrorList** cho việc thông báo lỗi hoặc cảnh báo trên một thông điệp trước đó. **CÓ THỂ** bỏ qua phần tử này;
- môđun an ninh (Security Module);
- **Signature** (chữ ký) – Một phần tử bao gồm một chữ kí số phù hợp với [XMLDSIG] đánh dấu dữ liệu liên kết với thông điệp. Phần tử này **CÓ THỂ** lược bỏ.

2.3.6 Nội dung phần tử #wildcard (Số đại diện)

Một số phần tử đuôi mở rộng SOAP của ebXML, như được chỉ trong giản đồ, cho phép mở rộng cho nội dung phần tử tên miền được hạn định nước ngoài. Nội dung phần tử đuôi mở rộng PHẢI là tên miền được hạn định theo XMLNS [XMLNS] và PHẢI thuộc một tên miền nước ngoài. Một tên miền nước ngoài KHÔNG PHẢI là http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd. Các phần tử đại diện được cung cấp khi giao thức có yêu cầu các đuôi mở rộng riêng hoặc các đuôi mở rộng trong tương lai.

Một sự thực thi MSH CÓ THỂ lược bỏ phần tử tên miền được hạn định và nội dung của nó.

2.3.7 Thuộc tính id (Nhận dạng)

Mỗi phần tử đuôi mở rộng SOAP của ebXML được định nghĩa trong tiêu chuẩn này có một thuộc tính *id* hoặc một ID XML giúp định danh duy nhất phần tử trong *thông điệp* SOAP. Nó CÓ THỂ được áp dụng cho chữ ký số của *thông điệp* SOAP của ebXML khi các phần tử đuôi mở rộng SOAP của ebXML riêng rẽ có thể được bao gồm hoặc loại trừ bằng một URI "#<idvalue>" (giá trị id) trong phần tử **Reference** (tham chiếu).

2.3.8 Thuộc tính version (Phiên bản)

Thuộc tính **version** (phiên bản) YÊU CẦU để chỉ rõ phiên bản quy định tiêu đề dịch vụ thông điệp ebXML mà các mở rộng tiêu đề SOAP của ebXML tuân theo. Mục đích là cung cấp thêm khả năng xác định phiên bản. Phù hợp với tiêu chuẩn này khi mọi thuộc tính **version** (phiên bản) trên bất kỳ phần tử đuôi mở rộng SOAP nào được quy định trong tiêu chuẩn này PHẢI có một giá trị là "2.0". Một thông điệp ebXML CÓ THỂ bao gồm các phần tử đuôi mở rộng tiêu đề SOAP có giá trị khác "2.0". Một sự thực thi phù hợp với tiêu chuẩn này nếu nó có khả năng định danh và xử lý một thông điệp với các mở rộng SOAP của ebXML có phiên bản khác "2.0". Nó PHẢI báo lỗi (chi tiết TBD) nếu nó không định danh được phiên bản. Thuộc tính **version** (phiên bản) PHẢI là tên miền hạn định cho tên miền các đuôi mở rộng **Envelope** SOAP (đường bao SOAP) của ebXML được quy định ở trên.

Việc sử dụng nhiều phiên bản của các phần tử đuôi mở rộng SOAP của ebXML trong và tài liệu SOAP của ebXML được hỗ trợ nhưng chỉ sử dụng các trường hợp giới hạn khi cần thiết cho việc thay đổi ngữ nghĩa Một phần tử mà không thể đợi lần phát hành phiên bản quy định dịch vụ thông điệp ebXML tiếp theo.

2.3.9 Thuộc tính mustUnderstand SOAP (Phải hiểu)

Thuộc tính **mustUnderstand** SOAP (phải hiểu) YÊU CẦU trên cơ sở các đuôi mở rộng **Header** (đuôi mở rộng) SOAP, tên miền hạn định cho tên miền SOAP (<http://shemas.xmlsoap.org/soap/envelope/>), chỉ rằng nội dung phần tử đó PHẢI được hiểu bởi một quá trình nhận hoặc quá trình khác, thông điệp còn lại PHẢI được loại bỏ phù hợp với SOAP [SOAP]. Thuộc tính này có giá trị '1' (đúng) chỉ phần tử

PHẢI được hiểu hoặc từ chối. Thuộc tính này có giá trị '0' (sai), mặc định, chỉ phần tử này có thể bị bỏ qua hoặc không được hiểu.

2.3.10 URI của chủ thể “MSH tiếp theo (Next MSH)” trong ebXML

URI *urn:oasis:names:tc:ebxml-msg:actor:nextMSH* khi được sử dụng trong nội dung giá trị thuộc tính **actor** SOAP (chủ thể SOAP) PHẢI được hiểu với nghĩa một thực thể có vai trò là một trường hợp MSH ebXML theo tiêu chuẩn này.

URI của **actor** (chủ thể) này cho phép các nút SOAP không PHẢI là các nút MSH ebXML có thể tham gia vào đường dẫn thông điệp của một thông điệp ebXML. Ví dụ một nút *thông điệp* SOAP được số hoá.

Mọi nút MSH ebXML PHẢI có vai trò này.

2.3.11 URI của chủ thể “To Party MSH” (MSH bên tham gia trước) trong ebXML

URL *urn: oasis:names:tc:ebxml-msg:actor:toPartyMSH* khi được sử dụng trong nội dung giá trị thuộc tính **actor** SOAP PHẢI được hiểu với nghĩa là một trường hợp của một nút MSH ebXML theo tiêu chuẩn này, có vai trò là bên tham gia được xác định trong phần tử **MessageHeader** (tiêu đề thông điệp)/**To/PartyId** của thông điệp. Một MSH ebXML có thể giữ vai trò này. cách thức thực hiện nằm ngoài phạm vi tiêu chuẩn này.

Đích cơ bản của thông điệp MSH ebXML PHẢI giữ vai trò của URI của chủ thể "To Party MSH" (MSH bên tham gia trước) trong phần bổ sung vai trò mặc định được quy định bởi SOAP.

3 Phần tử đuôi mở rộng lõi

3.1 Phần tử MessageHeader (Tiêu đề thông điệp)

Phần tử **MessageHeader** (tiêu đề thông điệp) là bắt buộc trong tất cả thông điệp ebXML. Nó PHẢI là Một phần tử con của phần tử **Header** SOAP.

Phần tử **MessageHeader** (tiêu đề thông điệp) là Một phần tử hỗn hợp được tạo nên từ các phần tử con sau:

- một thuộc tính **id** (xem chi tiết phần 2.3.7);
- một thuộc tính **version** (phiên bản) (xem chi tiết phần 2.3.8);
- một thuộc tính **mustUnderstand** SOAP (phải hiểu) có giá trị “1” (xem chi tiết phần 2.3.9);
- phần tử **From** (xuất phát từ);
- phần tử **To** (hướng đến);
- phần tử **CPAId** (id của CPA);
- phần tử **ConversationId** (id của hội thoại);
- phần tử **Service** (dịch vụ);

TCVN ISO/TS 15000-2 : 2007

- phần tử **Action** (hành động);
- phần tử **MessageData** (dữ liệu thông điệp);
- phần tử **DuplicateElimination** (loại trừ sao chép);
- phần tử **Description** (mô tả).

3.1.1 Phần tử **From** (xuất phát từ) và phần tử **To** (hướng đến)

Phần tử **From** (xuất phát từ) YÊU CẦU xác định bên khởi tạo thông điệp. Phần tử **To** (hướng đến) YÊU CẦU xác định bên nhận thông điệp. Cả hai phần tử **To** (hướng đến) và **From** (xuất phát từ) có thể bao gồm các định danh lôgic như một số hiệu DUNS hoặc các định danh vật lí như một địa chỉ email.

Mỗi phần tử **From** (xuất phát từ) và **To**(hướng đến) bao gồm:

- phần tử **PartyId** (id bên tham gia) – Xuất hiện một hoặc nhiều lần;
- phần tử **Role** (vai trò) – Xuất hiện không hoặc một lần.

Nếu phần tử **From** (xuất phát từ) hoặc **To** (hướng đến) bao gồm nhiều phần tử **PartyId** (id bên tham gia) thì tất cả các thành phần trong danh sách PHẢI được định danh trong cùng tổ chức. Trừ khi giá trị thuộc tính **type** (kiểu) đơn lẻ dùng cho nhiều hệ thống định danh, giá trị của bất kỳ thuộc tính **type** (kiểu) cho trước nào PHẢI là duy nhất trong danh sách các phần tử **PartyId** (id bên tham gia) trong phần tử **From** (xuất phát từ) hoặc **To** (hướng đến).

CHÚ THÍCH: Cơ chế này đặc biệt có lợi khi truyền thông điệp giữa các bên dùng đa phương tiện. Khái quát hơn, *From Party (Bên tham gia From)* cung cấp sự định danh trên tất cả các tên miền mà nó biết nhằm hỗ trợ trung gian và nơi đến thể ưu tiên các hệ thống định danh đặc biệt.

Các phần tử **From** (xuất phát từ) và **To** (hướng đến) bao gồm không hoặc Một phần tử con **Role** (vai trò) mà nếu tồn tại thì PHẢI theo ngay sau phần tử con **PartyId** (id bên tham gia) cuối cùng.

3.1.1.1 Phần tử **PartyId** (id bên tham gia)

Phần tử **PartyId** (id bên tham gia) có một thuộc tính **type** (kiểu) và nội dung là một chuỗi giá trị. Thuộc tính **type** (kiểu) chỉ thị vùng của các tên mà thuộc chuỗi trong nội dung phần tử **PartyId** (id bên tham gia). Giá trị của thuộc tính **type** (kiểu) PHẢI được thoả thuận lẫn nhau và các bên đều hiểu được. Giá trị thuộc tính **type** (kiểu) được KHUYẾN CÁO là một URI. Các giá trị này còn được KHUYẾN CÁO sử dụng các tài liệu EDIRA (ISO 6523), EDIFACT ISO 9735 hoặc ANSI ASC X12 105.

Nếu thuộc tính **type** (kiểu) của **PartyId** (id bên tham gia) không tồn tại thì nội dung phần tử **PartyId** (id bên tham gia) PHẢI là một URI [RFC2396], ngược lại MSH nhận PHẢI thông báo một lỗi (xem phần 2.1.5) với **errorCode** (mã lỗi) là **inconsistent** và **severity** là **Error**. Nội dung phần tử **PartyId** (id bên tham gia) được KHUYẾN CÁO là một URI.

3.1.1.2 Phần tử Role (Vai trò)

Phần tử **Role** (vai trò) xác nhận vai trò được phép (**fromAuthorizedRole** (vai trò được phép xuất phát từ) hoặc **toAuthorizeRole** (vai trò được phép hướng đến)) của bên gửi (nếu tồn tại Một phần tử con của phần tử **From** (xuất phát từ)) và/hoặc của bên nhận (nếu tồn tại Một phần tử con của phần tử **To** (hướng đến)) thông điệp. Giá trị của phần tử **Role** (vai trò) là một chuỗi không rỗng và được quy định trong CPA.

CHÚ THÍCH: Vai trò nên là một URI – Ví dụ: <http://rosettanet.org/roles/buyer>.

Đoạn sau minh họa việc sử dụng phần tử **From** (xuất phát từ) và **To** (hướng đến).

```
<eb:From>
  <eb:PartyId eb:type="urn:duns">123456789</eb:PartyId>
  <eb:PartyId eb:type="SCAC">RDWY</PartyId>
  <eb:Role>http://rosettanet.org/roles/Buyer</eb:Role>
</eb:From>
<eb:To>
  <eb:PartyId>mailto:joe@example.com</eb:PartyId>
  <eb:Role>http://rosettanet.org/roles/Seller</eb:Role>
</eb:To>
```

3.1.2 Phần tử CPAId (id của CPA)

Phần tử **CPAId** (id của CPA) YÊU CẦU là một chuỗi tham số chủ yếu của thông điệp trao đổi giữa các bên. Bên nhận thông điệp PHẢI có thể phân tích CPAId (id của CPA) thành một tập tham số riêng trong trường mục người gửi thông điệp.

Giá trị của Một phần tử **CPAId** (id của CPA) PHẢI là duy nhất trong một tên miền được hai bên thỏa thuận. Đây là một sự móc nối các giá trị **partyId** (id của bên tham gia) của phần tử **From** và **To**, một URI được đặt trước bằng tên miền internet của một trong các bên hoặc một tên miền được đưa ra và quản lý bằng một số dịch vụ đăng ký hoặc đặt tên khác. CPAId (id của CPA) được KHUYẾN CÁO là một URI.

CPAId (id của CPA) CÓ THỂ tham chiếu một trường hợp của một CPA được quy định trong Sơ lược giao thức cộng tác ebXML và Đặc tả thỏa thuận [ebCPP]. Một ví dụ về phần tử **CPAId** (id của CPA) như sau:

```
<eb:CPAId>http://example.com/cpas/ourcpawithyou.xml</eb:CPAId>
```

Các tham số thông điệp được xác định bằng các phần tử thích hợp từ CPA được xác định bằng phần tử **CPAId** (id của CPA).

Nếu một người nhận xác định một thông điệp xung đột với CPA thì trình điều khiển thích hợp của sự xung đột này là không được quy định trong tiêu chuẩn này. Do đó, người gửi KHÔNG NÊN tạo ra thông điệp trừ khi họ biết trước người nhận có khả năng giải quyết sự xung đột này.

Nếu một MSH nhận tìm thấy một sự mâu thuẫn thì nó PHẢI thông báo với một **errorCode** (mã lỗi) là **inconsistent** (mâu thuẫn) và **severity** (tính quy định) là **Error** (lỗi). Nếu không nhận ra **CPAId** (id của

TCVN ISO/TS 15000-2 : 2007

CPA) thì nó PHẢI thông báo với một **errorCode** (mã lỗi) là **notRecognized** (không công nhận) và một **severity** (tính quy định) là **Error** (lỗi).

3.1.3 Phần tử **ConversationId** (id của hội thoại)

Phần tử **ConversationId** (id của hội thoại) YÊU CẦU là một chuỗi định danh các thông điệp liên quan đánh dấu một hội thoại giữa hai bên. Nó PHẢI là duy nhất trong nội dung của **CPAId** (id của CPA) được quy định. Bên khởi tạo một hội thoại xác định giá trị của phần tử **ConversationId** (id của hội thoại) PHẢI được phản hồi trong mọi thông điệp về hội thoại.

ConversationId (id của hội thoại) cho phép bên nhận thông điệp xác nhận tình trạng của một ứng dụng hoặc quá trình sinh ra hoặc điều khiển các thông điệp trước đó trong một hội thoại. Nó giữ sự liên tục cho mọi thông điệp trong một hội thoại.

Giá trị được sử dụng cho một **ConversationId** (id của hội thoại) phụ thuộc vào việc thực thi. Một ví dụ về phần tử **ConversationId** (id của hội thoại) như sau:

```
<eb:ConversationId>20001209-133003-28572</eb:ConversationId>
```

CHÚ THÍCH: Việc thực thi tùy ý chọn cách định danh và lưu trữ trạng thái hội thoại liên quan đến một hội thoại cụ thể. Phần mềm thực thi PHẢI cung cấp sự hài hoà cho việc ánh xạ giữa giản đồ xác minh của họ và một **ConversationId** (id của hội thoại) được tạo ra bởi một sự thực thi khác.

3.1.4 Phần tử **Service** (Dịch vụ)

Phần tử **Service** (dịch vụ) yêu cầu xác định dịch vụ tác động đến thông điệp và nó được quy định bởi người thiết kế dịch vụ. Người thiết kế dịch vụ có thể là:

- một tổ chức tiêu chuẩn hoá hoặc;
- một cá nhân hoặc xí nghiệp.

CHÚ THÍCH: Trong nội dung của một kiểu quá trình thương mại ebXML, một hành động thấp nhất có thể đóng vai trò hoạt động cơ bản trong quá trình thương mại [ebBPSS] (yêu cầu hoặc từ chối vai trò) và một dịch vụ là một tập các hành động liên quan cho một vai trò cho phép trong một bên tham gia.

Một ví dụ về phần tử **Service** (dịch vụ) như sau:

```
<eb:Service>urn:services:SupplierOrderProcessing</eb:Service>
```

CHÚ THÍCH: URI trong Phần tử **Service** (dịch vụ) bắt đầu bằng tên miền **urn:oasis:names:tc:ebxml-msg:service** được tiêu chuẩn này dành sẵn cho việc sử dụng.

Phần tử **Service** (dịch vụ) có một thuộc tính **type** (kiểu) đơn.

3.1.4.1 Thuộc tính **type** (kiểu)

Nếu thuộc tính **type** (kiểu) xuất hiện, thì nó chỉ cho các bên gửi và bên nhận thông điệp biết cách giải thích nội dung của phần tử **Service** (dịch vụ). Hai phần này có thể sử dụng giá trị của thuộc tính **type** (kiểu) để hỗ trợ việc giải thích.

Nếu thuộc tính **type** (kiểu) không xuất hiện, thì nội dung của phần tử **Service** (dịch vụ) PHẢI là URI [RFC2396]. Nếu nó không PHẢI là URI thì thông báo một lỗi với **errorCode** (mã lỗi) là **Inconsistent** và **severity** (tính nghiêm ngặt) là **Error** (xem phần 4.1.5)

3.1.5 Phần tử **Action** (Hành động)

Phần tử **Action** (hành động) yêu cầu định danh một quá trình trong khi **Service** (dịch vụ) xử lý thông điệp. Phần tử **Action** (hành động) là duy nhất trong phần tử **Service** (dịch vụ) mà nó được xác định. Giá trị của phần tử **Action** (hành động) được quy định bởi trình thiết kế dịch vụ. Dưới đây là ví dụ về phần tử **Action** (hành động).

```
<eb:Action>NewOrder</eb:Action>
```

Nếu giá trị của phần tử **Action** (hành động) hoặc phần tử **Service** (dịch vụ) không được chấp nhận bởi **MSH** nhận, thì nó PHẢI thông báo lỗi với **errorCode** (mã lỗi) là **NotRecognized** (không công nhận) và **severity** là **Error**.

3.1.6 Phần tử **MessageData** (Dữ liệu thông điệp)

Phần tử **MessageData** (dữ liệu thông điệp) YÊU CẦU cung cấp một phương thức định danh duy nhất thông điệp ebXML. Nó bao gồm:

- phần tử **MessageId** (id của thông điệp);
- phần tử **Timestamp** (tem thời gian);
- phần tử **RefToMessageId** (id của thông điệp tham chiếu đến);
- phần tử **TimeToLive** (thời gian làm việc).

Dưới đây là đoạn minh họa cấu trúc của phần tử **MessageData** (dữ liệu thông điệp):

```
<eb:MessageData>
  <eb:MessageId>20001209-133003-28572@example.com</eb:MessageId>
  <eb:Timestamp>2001-02-15T11:12:12</eb:Timestamp>
  <eb:RefToMessageId>20001209-133003-28571@example.com</eb:RefToMessageId>
</eb:MessageData>
```

3.1.6.1 Phần tử **MessageId** (id của thông điệp)

Phần tử **MessageId** (id của thông điệp) YÊU CẦU là từ định danh duy nhất mang tính tổng thể cho mỗi thông điệp phù hợp với **MessageId** (id của thông điệp) [RFC2822].

CHÚ THÍCH: Trong các tiêu đề **Message-Id** (ID-Thông điệp) và **Content - Id** (ID-Nội dung) của MIME, các giá trị luôn luôn được đặt trong dấu ngoặc đơn. Tuy nhiên, dấu chỉ dẫn tham khảo đoạn ở mid: hoặc cid: gián đồ của URI và các phần tử **MessageId** (id của thông điệp) và **RefToMessageId** PHẢI không được bao gồm các dấu phân cách này.

3.1.6.2 Phần tử **Timestamp** (Tem thời gian)

Phần tử yêu cầu **Timestamp** (tem thời gian) là giá trị biểu trưng cho thời gian mà tiêu đề thông điệp được tạo nên phù hợp với `dateTime` [XMLSchema (giản đồ XML)] và PHẢI được thể hiện giống như UTC. Việc biểu thị UTC trong phần tử **Timestamp** (tem thời gian) bằng từ định danh "Z" là tùy ý.

3.1.6.3 Phần tử **RefToMessageId** (id của thông điệp tham chiếu đến)

Phần tử **RefToMessageId** (id của thông điệp tham chiếu đến) có một tập các yếu tố 0 hoặc 1. Khi xuất hiện, nó PHẢI chứa giá trị **MessageId** (id của thông điệp) của thông điệp ebXML ban đầu mà thông điệp này có liên quan. Nếu không có thông điệp có liên quan ban đầu, thì phần tử này không được xuất hiện.

Đối với các thông điệp Error (lỗi), phần tử **RefToMessageId** (id của thông điệp tham chiếu đến) được YÊU CẦU và giá trị của nó PHẢI là giá trị **MessageId** (id của thông điệp) của thông điệp lỗi (như chỉ ra trong phần 4.2).

3.1.6.4 Phần tử **TimeToLive** (Thời gian làm việc)

Nếu phần tử **TimeToLive** (thời gian làm việc) xuất hiện, thì nó PHẢI được dùng để chỉ thời gian, (biểu thị giống như UTC), bằng thông điệp cần được phát đi tới *To Party MSH*. Nó PHẢI phù hợp với Giản đồ `dateTime` của XML.

Trong trường hợp này, **TimeToLive** (thời gian làm việc) hết hiệu lực nếu thời gian của đồng hồ nội bộ điều chỉnh so với UTC của *MSH nhận* là lớn hơn giá trị của **TimeToLive** của thông điệp.

Nếu MSH của bên tham gia đến (To Party) nhận được thông điệp tại nơi mà **TimeToLive** (thời gian làm việc) đã hết hiệu lực, thì nó gửi một thông điệp tới MSH của bên tham gia đến từ (From Party), thông báo rằng **TimeToLive** (thời gian làm việc) của thông điệp đã hết hiệu lực. Thông điệp này được bao gồm một **ErrorList** (danh sách lỗi) chứa một lỗi với thuộc tính **errorCode** (mã lỗi) thiết lập là **TimeToLiveExpired** và thuộc tính **severity** thiết lập là **Error**.

Phần tử **TimeToLive** (thời gian làm việc) được thảo luận thêm theo việc truyền thông điệp tin cậy (thông điệp xác thực) trong phần 6.4.5.

3.1.7 Phần tử **DuplicateElimination** (Loại trừ sao chép)

Nếu xuất hiện, phần tử **DuplicateElimination** (loại trừ sao chép) định danh yêu cầu của người gửi cho MSH nhận để kiểm tra bản sao thông điệp (xem chi tiết phần 6.4.1).

Các giá trị hợp lệ cho **DuplicateElimination** (loại trừ sao chép):

- **DuplicateElimination** (loại trừ sao chép) xuất hiện - bản sao thông điệp cần được loại bỏ;
- **DuplicateElimination** (loại trừ sao chép) không xuất hiện - các kết quả này phát đi trong hoạt động của Best - Effort.

Phần tử **DuplicateElimination** (loại trừ sao chép) không được xuất hiện nếu CPA có **duplicateElimination** (loại trừ sao chép) thiết lập là **never** (xem chi tiết phần 6.4.1 và phần 6.6).

3.1.8 Phần tử Description (Mô tả)

Phần tử **Description** (mô tả) có thể không xuất hiện hoặc xuất hiện nhiều lần. Mục đích của nó là giúp có thể đọc được sự diễn tả ý nghĩa và mục đích của thông điệp. Ngôn ngữ của việc diễn tả được xác định bởi thuộc tính yêu cầu **xml:lang**. Thuộc tính **xml:lang** PHẢI tuân theo các quy tắc của ngôn ngữ định danh trên lý thuyết trong XML [XML]. Mỗi sự kiện nên có một giá trị khác nhau cho **xml:lang**.

3.1.9 Ví dụ tiêu biểu về MessageHeader (Tiêu đề thông điệp)

Các đoạn dưới đây minh họa cấu trúc của phần tử **MessageHeader** (tiêu đề thông điệp) trong SOAP Header (tiêu đề SOAP).

```
<eb:MessageHeader eb:id="..." eb:version="2.0"
SOAP:mustUnderstand="1">
  <eb:From>
    <eb:PartyId>uri:example.com</eb:PartyId>
    <eb:Role>http://rosettanet.org/roles/Buyer</eb:Role>
  </eb:From>
  <eb:To>
    <eb:PartyId eb:type="someType">QRS543</eb:PartyId>
    <eb:Role>http://rosettanet.org/roles/Seller</eb:Role>
  </eb:To>
  <eb:CPAId>http://www.oasis-open.org/cpa/123456</eb:CPAId>
  <eb:ConversationId>987654321</eb:ConversationId>
  <eb:Service eb:type="myservicetypes">QuoteToCollect</eb:Service>
  <eb:Action>NewPurchaseOrder</eb:Action>
  <eb:MessageData>
    <eb:MessageId>UUID-2</eb:MessageId>
    <eb:Timestamp>2000-07-25T12:19:05</eb:Timestamp>
    <eb:RefToMessageId>UUID-1</eb:RefToMessageId>
  </eb:MessageData>
  <eb:DuplicateElimination/>
</eb:MessageHeader>
```

3.2 Phần tử Manifest (Bảng kê)

Phần tử **Manifest** (bảng kê) có thể xuất hiện giống như Một phần tử con của phần tử **Body** (thân) của SOAP (nội dung chính của SOAP). Phần tử **Manifest** (bảng kê) là Một phần tử hỗn hợp bao gồm một hoặc nhiều phần tử **Reference** (tham chiếu). Mỗi phần tử **Reference** (tham chiếu) định danh dữ liệu vùng mang thông tin tương ứng với thông điệp, bao gồm một phần của thông điệp bằng tài liệu vùng mang thông tin chứa trong phần chứa vùng mang thông tin hoặc các nguồn biệt lập có thể được sử dụng thông qua URL. Nó được đòi hỏi rằng không có dữ liệu vùng mang thông tin nào được xuất hiện trong **Body** (thân) của SOAP. Mục đích của **Manifest** (bảng kê) là:

- dễ dàng rút trực tiếp vùng mang thông tin riêng liên quan thông điệp ebXML;
- cho phép ứng dụng xác định rõ xem nó có thể xử lý vùng mang thông tin mà không cần PHẢI phân tách nó hay không.

Phần tử **Manifest** gồm có:

TCVN ISO/TS 15000-2 : 2007

- một thuộc tính **id** (xem chi tiết phần 2.3.7);
- một thuộc tính **version** (phiên bản) (xem chi tiết phần 2.3.8);
- một hoặc nhiều phần tử **Reference** (tham chiếu).

3.2.1 Phần tử Reference (Tham chiếu)

Phần tử **Reference** (tham chiếu) là Một phần tử hỗn hợp bao gồm các phần tử phụ thuộc dưới đây:

- không hoặc nhiều phần tử **Schema** (giản đồ) – Thông tin về giản đồ xác định trường hợp tài liệu đã định danh trong phần tử **Reference** (tham chiếu) nguồn;
- không hoặc nhiều Phần tử **Description** (mô tả) – Phần mô tả nguyên văn của đối tượng mang thông tin đã tham khảo bằng phần tử **Reference** (tham chiếu) nguồn.

Bản thân phần tử **Reference** (tham chiếu) là một kết nối đơn [XLINK]. Nó cần được lưu ý rằng việc sử dụng XLINK trong ngữ cảnh này được lựa chọn duy nhất cho mục đích cung cấp vốn từ vựng ngắn gọn để mô tả sự kết hợp. Việc sử dụng bộ xử lý XLINK hoặc động cơ là KHÔNG YÊU CẦU, nhưng có thể chứng minh một cách hữu ích trong các việc thực thi nào đó.

Phần tử **Reference** (tham chiếu) có nội dung thuộc tính dưới đây bổ sung cho nội dung phần tử đã mô tả ở trên:

- **id** – ID XML cho phần tử **Reference** (tham chiếu);
- **xlink:type** – Thuộc tính này định rõ phần tử giống như sự kết nối đơn XLINK. Nó có một giá trị cố định là 'simple';
- **xlink:href** – Thuộc tính YÊU CẦU này có một giá trị là URI của đối tượng mang thông tin đã tham khảo. Nó PHẢI phù hợp với tiêu chuẩn kỹ thuật XLINK [XLINK] cho kết nối đơn;
- **xlink:role** – Thuộc tính này định danh một số nguồn mô tả đối tượng mang thông tin hoặc mục đích của nó. Nếu xuất hiện, thì nó PHẢI có giá trị URI hợp lý phù hợp với quy định kỹ thuật [XLINK];
- bất cứ thuộc tính tên miền được hạn định nào khác đều CÓ THỂ xuất hiện. MSH nhận CÓ THỂ chọn dùng để bỏ qua bất cứ thuộc tính tên miền ngẫu nhiên nào khác với vấn đề đã xác định ở trên.

Trình thiết kế quá trình giao dịch hoặc trao đổi thông tin sử dụng ebXML (thông điệp ebXML) dữ liệu vùng mang thông tin được tham khảo bởi **Manifest** (bảng kê) và các giá trị được sử dụng cho **xlink:role**.

3.2.1.1 Phần tử Schema (Giản đồ)

Nếu mục chọn được tham khảo có một số giản đồ mô tả nó (ví dụ giản đồ XML, DTD và hoặc giản đồ cơ sở dữ liệu), thì phần tử **Schema** (giản đồ) nên được xuất hiện giống như Một phần tử con của phần tử **Reference** (tham chiếu). Nó cung cấp phương pháp định danh giản đồ và phiên bản của nó hạn chế

nội dung đối tượng mang thông tin đã định danh bởi phần tử **Reference** (tham chiếu) gốc. Phần tử **Schema** (giản đồ) bao gồm các thuộc tính sau:

- **location** (vị trí) – URI yêu cầu của giản đồ;
- **version** (phiên bản) – Phiên bản từ định danh của giản đồ.

3.2.1.2 Phần tử Description (Mô tả)

Xem chi tiết phần 3.1.8. Dưới đây là ví dụ của phần tử **Description** (mô tả)

```
<eb:Description xml:lang="en-GB">Purchase Order for 100,000
widgets</eb:Description>
```

3.2.2 Hiệu lực Manifest (bảng kê)

Nếu thuộc tính **xlink:href** bao gồm URI là id của nội dung (giản đồ URI "cid") thì phần MIME với `content-id` đó PHẢI được xuất hiện trong *phần chứa vùng mang thông tin* (Payload Container) tương ứng của thông điệp. Nếu không PHẢI vậy thì lỗi được thông báo tới *bên tham gia From* (From Party) với **errorCode** (mã lỗi) là **MimeProblem** và **severity** là **Error**.

Nếu thuộc tính **xlink:href** bao gồm URI không là id của nội dung (giản đồ URI "cid") và URI không thể được giải quyết thì nó là một quyết định thực thi xem nó có báo cáo lỗi hoặc không. Nếu lỗi được báo cáo, nó PHẢI được báo cáo tới *From Party* với **errorCode** (mã lỗi) là **MimeProblem** và **severity** là **Error**.

CHÚ THÍCH: Nếu việc vùng mang thông tin tồn tại mà không được tham khảo bởi Manifest, thì việc vùng mang thông tin đó NÊN được loại bỏ.

3.2.3 Ví dụ về Manifest (bảng kê)

Đoạn sau minh họa một **Manifest** điển hình về phần nội dung chính MIME vùng mang thông tin đơn:

```
<eb:Manifest eb:id="Manifest" eb:version="2.0"> 994
  <eb:Reference eb:id="pay01" 995
    xlink:href="cid:payload-1" 996
    xlink:role="http://regrep.org/gci/purchaseOrder"> 997
  <eb:Schema
    eb:location="http://regrep.org/gci/purchaseOrder/po.xsd"
    eb:version="2.0"/> 998
    <eb:Description xml:lang="en-US">Purchase Order for 100,000
    widgets</eb:Description> 999
  </eb:Reference> 1000
</eb:Manifest> 1001
```

4 Các môđun lỗi

4.1 Môđun an ninh

Một *dịch vụ thông điệp XML* đưa ra có rủi ro an ninh. Một *dịch vụ thông điệp* có thể chịu rủi ro bởi:

- truy cập trái phép;
- sự tấn công tính bảo mật và/hoặc toàn vẹn dữ liệu (ví dụ sự tấn công dữ liệu thông qua các người xử lý dữ liệu ở giai đoạn giữa);

– sự phủ nhận dịch vụ và giả mạo.

Mỗi rủi ro an ninh được nói chi tiết ở mục ebXML Technical Architecture Risk Assessment Technical Report [secRISK]. (Báo cáo kỹ thuật về đánh giá rủi ro kiến trúc kỹ thuật ebXML)

Mỗi rủi ro an ninh này được hướng vào toàn bộ hoặc từng phần tùy theo ứng dụng của nó hoặc có sự kết hợp của biện pháp đối phó được nói đến ở mục này. Tiêu chuẩn này mô tả một bộ các mô tả hoặc kết hợp của biện pháp đối phó đã lựa chọn để chỉ ra các rủi ro chính dựa trên các kỹ thuật có sẵn được dùng phổ biến. Mỗi mô tả theo lý thuyết bao gồm việc mô tả các nguy hiểm không được hướng vào. Xem phụ lục C về bảng hồ sơ an ninh.

Sự áp dụng biện pháp đối phó NÊN cân đối giữa rủi ro cố hữu và giá trị tài sản có thể chịu rủi ro. Trong tiêu chuẩn này, một thông điệp *được ký* thông điệp nào đó bao gồm phần tử **Signature**.

4.1.1 Phần tử Signature (Chữ ký)

Một thông điệp ebXML có thể được ký số để đưa ra biện pháp an ninh. Không hoặc nhiều các phần tử **Signature** (chữ ký) thuộc chữ ký XML [XMLDSIG] tên miền xác định, có thể tồn tại như phần tử con của phần tử **Header** SOAP. Phần tử **Signature** (chữ ký) PHẢI là tên miền được hạn định phù hợp với chữ ký XML [XMLDSIG]. Nội dung và cấu trúc của phần tử **Signature** (chữ ký) PHẢI phù hợp với quy định kỹ thuật của chữ ký XML [XMLDSIG]. Nếu có nhiều hơn Một phần tử **Signature** (chữ ký) trong Tiêu đề SOAP thì phần tử đầu tiên PHẢI tương ứng với chữ ký số của thông điệp ebXML như được ký bởi *MSH của bên tham gia đến từ (From Party)* phù hợp với mục 4.1. Phần tử **Signature** (chữ ký) thêm vào có thể là kết quả của nó thì không được xác định bằng tiêu chuẩn này.

Xem mục 4.1.3 chi tiết về cấu trúc của phần tử **Signature** (chữ ký) khi ký số một thông điệp ebXML.

4.1.2 Quản lý và an ninh

Không có một công nghệ nào (cho dù nó tiên tiến như thế nào) là thích hợp để thay thế cho việc áp dụng có hiệu quả các thực tiễn và chính sách quản lý an ninh.

Cần KHUYẾN CÁO vị trí quản lý của *dịch vụ thông điệp ebXML* có hiệu quả do việc áp dụng đối với việc quản lý và cung cấp các kỹ thuật An ninh, vị trí của các quy trình An ninh, các giao thức mã hoá, sự bổ xung cập nhật và việc ứng dụng PHẢI được sắp xếp một cách thích hợp. (Xem <http://www.cert.org/> và <http://ciac.llnl.gov/>).

4.1.2.1 Thoả thuận giao thức cộng tác

Cấu hình an ninh của các MSH được ghi rõ trong CPA. Hai phạm vi CPA có các định nghĩa an ninh như sau:

- Document Exchange (trao đổi tài liệu) áp dụng an ninh cho vùng mang thông tin của thông điệp. MSH không chịu trách nhiệm về bất kỳ an ninh nào ở mức này nhưng có thể cung cấp các dịch vụ này cho bên gửi thông điệp;

- Transport (truyền tải) áp dụng an ninh cho toàn bộ tài liệu ebXML, bao gồm tiêu đề và vùng mang thông tin.

4.1.3 Tạo chữ ký

Một thông điệp ebXML được ký sử dụng XMLDSIG theo các bước sau đây:

1. tạo phần tử **SignedInfo** (thông tin được ký) cùng với với các phần tử **SignatureMethod** (phương pháp ký), **CanonicalizationMethod** (phương pháp chuẩn) và **Reference** (tham chiếu) cho **Envelope** SOAP (đường bao SOAP) (đường bao SOAP) và bất kỳ đối tượng mang thông tin nào theo quy định chữ ký XML [XMLDSIG] (XMLDSIG);
2. chuẩn hoá và sau đó tính toán **SignatureValue** (giá trị ký) thông qua **SignedInfo** (thông tin được ký) trên cơ sở các thuật toán được quy định trong **SignedInfo** (thông tin được ký) như trong chữ ký XML [XMLDSIG];
3. xây dựng phần tử **Signature** (chữ ký) bao gồm các phần tử **SignedInfo** (thông tin được ký), **keyInfo** (KHUYẾN CÁO) và **SignatureValue** (giá trị ký) như trong chữ ký XML [XMLDSIG];
4. bao gồm phần tử **Signature** tên miền được hạn định trong Tiêu đề SOAP vừa ký.

Phần tử **SignedInfo** (thông tin được ký) PHẢI có Một phần tử **CanonicalizationMethod** (phương pháp chuẩn), Một phần tử **SignatureMethod** (phương pháp ký) và một hoặc nhiều phần tử **Reference** (tham chiếu) như trong chữ ký XML [XMLDSIG].

KHUYẾN CÁO về phương pháp chuẩn hoá được áp dụng cho dữ liệu được ký hiệu như sau:

```
<CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
```

được mô tả trong [XMLC14N]. Thuật toán này không gồm chú thích.

Phần tử **SignatureMethod** PHẢI có mặt và có một thuộc tính **Algorithm** (thuật toán).

Giá trị KHUYẾN CÁO cho thuộc tính **Algorithm** (thuật toán) là:

```
<SignatureMethod Algorithm="http://www.w3.org/2000/09/xmlsig#dsa-sha1"/>
```

Giá trị KHUYẾN CÁO này PHẢI được hỗ trợ bởi tất cả phần mềm thực thi *dịch vụ thông điệp ebXML* phù hợp.

Phần tử [XMLDSIG] **Reference** của tài liệu **Envelope** SOAP (đường bao SOAP) PHẢI có một giá trị thuộc tính URI là "" để tạo ra chữ ký được dùng trong tài liệu bao gồm phần tử **Signature**.

Phần tử [XMLDSIG] **Reference** của **Envelope** SOAP (đường bao SOAP) có thể bao gồm thuộc tính **type** (kiểu) giá trị "<http://www.w3.org/2000/09/xmlsig#Object>" phù hợp với XMLDSIG. Thuộc tính này chỉ bổ sung kiến thức. Nó có thể bị bỏ qua. Sự bổ sung MSH ebXML PHẢI được sẵn sàng để vận dụng trong các trường hợp khác. Phần tử **Reference** (tham chiếu) có thể gồm thuộc tính **id**.

TCVN ISO/TS 15000-2 : 2007

Phần tử [XMLDSIG] **Reference** (tham chiếu) của **Envelope** SOAP (đường bao SOAP) PHẢI gồm Một phần tử **Transforms** (truyền tải) con. Phần tử **Transforms** (truyền tải) PHẢI gồm các phần tử con **Transforms** (truyền tải) sau đây:

- Phần tử **Transform** (truyền tải) đầu tiên có một thuộc tính **Algorithm** (thuật toán) với giá trị:

```
<Transform Algorithm="http://www.w3.org/2000/09/xmlsig#enveloped-signature"/>
```

Kết quả của câu lệnh này không gồm phần tử **Signature** (chữ ký) gốc và tất cả các phần tử con của nó

- Phần tử **Transform** (truyền tải) thứ hai có Một phần tử **XPath** con có giá trị:

```
<Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
<XPath> not (ancestor-or-self::node() [@SOAP:actor="urn:oasis:names:tc:ebxml-
msg:actor:nextMSH"]
ancestor-or-self::node() [@SOAP:actor="http://schemas.xmlsoap.org/soap/actor/next"] )
</XPath>
</Transform>
```

Kết quả của câu lệnh [XPath] này không gồm tất cả các phần tử có trong **Envelope** SOAP (đường bao SOAP) chứa một SOAP: thuộc tính **actor** nhằm tới **nextMSH** và tất cả các phần tử con của nó. Nó cũng không gồm tất cả các phần tử có thuộc tính **actor** nhằm tới phần tử ở nút tiếp theo (có thể thay đổi ở cuối tuyến - route). Tất cả nút trung gian hoặc MSH PHẢI không được thay đổi, định dạng hoặc thay đổi theo bất cứ cách nào của bất cứ các phần tử không hướng tới nút trung gian. Các nút trung gian không PHẢI thêm hoặc xoá bớt khoảng cách trắng. Các thay đổi như vậy có thể làm mất hiệu lực chữ ký.

Phần tử **Transform** (truyền tải) cuối và nên có một thuộc tính **Algorithm** (thuật toán) với giá trị:

```
<Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
```

Kết quả của thuật toán là để chuẩn hóa **Envelope** SOAP (đường bao SOAP) XML và không gồm các chú thích.

CHÚ THÍCH: Các biến đổi này được dành cho **Envelope** SOAP (đường bao SOAP) và nội dung của nó. Các biến đổi này không dành cho các đối tượng mang thông tin. Sự quyết định các biến đổi thích hợp đối với mỗi vùng mang thông tin được để lại để bổ xung.

Mỗi đối tượng mang thông tin yêu cầu việc ký PHẢI được trình bày bằng Một phần tử [XMLDSIG] **Reference** thuộc tính **URI** quyết định đối tượng mang thông tin. Điều này có thể là **Content-Id** URI của phần thân MIME của đối tượng mang thông tin mà cũng có thể là một URI phù hợp **Content-Location** của phần thân MIME của đối tượng mang thông tin hoặc có thể là một URI quyết định đối tượng mang thông tin ngoài *gói thông điệp (Message Package)*. KHUYẾN CÁO là giá trị thuộc tính URI tương đương với giá trị URI **xlink:href** của phần tử tương ứng **Manifest/Reference** cho đối tượng mang thông tin.

CHÚ THÍCH: Khi một mã hóa đường truyền (ví dụ base64) được quy định bởi một tiêu đề Content-Transfer-Encoding MIME được dùng cho **Envelope** SOAP (đường bao SOAP) hoặc các đối tượng mang thông tin thì việc tạo ra chữ ký phải được thực hiện trước khi mã hóa.

Ví dụ về thông điệp được ký số SOAP của ebXML:

```
<?xml version="1.0" encoding="utf-8"?>
<SOAP:Envelope xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-
2_0.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
  http://www.oasis-open.org/committees/ebxml-
msg/schema/envelope.xsd
  http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
header-2_0.xsd
  http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
header-2_0.xsd">
  <SOAP:Header>
    <eb:MessageHeader eb:id="..." eb:version="2.0" SOAP:mustUnderstand="1">
      ...
    </eb:MessageHeader>
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
      <SignedInfo>
        <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
        <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
        <Reference URI="">
          <Transforms>
            <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature"/>
            <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
              <XPath> not (ancestor-or-self::node() [@SOAP:actor=
                &quot;urn:oasis:names:tc:ebxml-msg:actor:nextMSH&quot;;]
                | ancestor-or-self::node() [@SOAP:actor=
                &quot;http://schemas.xmlsoap.org/soap/actor/next&quot;;])
            </XPath>
          </Transform>
          <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <DigestValue>...</DigestValue>
      </Reference>
      <Reference URI="cid://blahblahblah/">
        <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <DigestValue>...</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>...</SignatureValue>
    <KeyInfo>...</KeyInfo>
  </Signature>
</SOAP:Header>
<SOAP:Body>
  <eb:Manifest eb:id="Mani01" eb:version="2.0">
    <eb:Reference xlink:href="cid://blahblahblah/"
xlink:role="http://ebxml.org/gci/invoice">
      <eb:Schema eb:version="2.0"
eb:location="http://ebxml.org/gci/busdocs/invoice.dtd"/>
    </eb:Reference>
  </eb:Manifest>
</SOAP:Body>
</SOAP:Envelope>
```

4.1.4 Các công nghệ đối phó

4.1.4.1 Chữ ký số dài hạn

Công nghệ sẵn dùng cho một thông điệp được ký số ebXML (gồm **Header** (tiêu đề), **Body** (thân) của SOAP ebXML và kết hợp các đối tượng mang thông tin của nó) mới được đưa ra dùng bằng một công nghệ phù hợp với W3C/IETF kết hợp với đặc tính chữ ký XML [XMLDSIG]. Một chữ ký XML phù hợp với đặc tính này có thể ký các phần chia của một tài liệu XML một cách có lựa chọn, cho phép các tài liệu được tăng lên (nội dung phần tử mới được thêm vào) trong khi vẫn duy trì được giá trị của (các) chữ ký.

Nếu các chữ ký được dùng để ký số thông điệp ebXML thì chữ ký XML [DSIG] PHẢI được dùng để gộp ebXML **Header** (tiêu đề) và **Body** (thân) SOAP với (các) phần chứa vùng mang thông tin ebXML hoặc dữ liệu ở một vùng khác trên web có liên quan tới thông điệp.

Một thông điệp ebXML yêu cầu một chữ ký số PHẢI được ký theo quy trình xác định yêu cầu kỹ thuật và PHẢI được tuân theo hoàn toàn chữ ký XML [XMLDSIG].

4.1.4.2 Việc nhận được ký dài hạn

Một thông điệp ebXML đã ký số CÓ THỂ được chấp nhận với một *thông điệp báo nhận* mà bản thân thông điệp này được ký số ở dạng như đã mô tả ở phần trước. *Thông điệp báo nhận* PHẢI gồm một danh sách các phần tử **Reference** (tham chiếu) [XMLDSIG] phù hợp với các phần tử được bao gồm trong phần tử **Signature** (chữ lý) của thông điệp ban đầu

4.1.4.3 Xác thực ngắn hạn

Xác thực ngắn hạn được cung cấp bởi kênh truyền thông được dùng để truyền Thông điệp ebXML. Việc xác thực này CÓ THỂ một chiều, hai chiều. Phương pháp cụ thể được xác định bởi giao thức truyền thông được dùng. Ví dụ, việc dùng một giao thức an ninh mạng, chẳng hạn như TLS [RFC2246] hoặc IPSEC [RFC2402] cấp cho bên gửi thông điệp ebXML một phương pháp xác thực với bên nhận trong môi trường TCP/IP.

4.1.4.4 Tính toàn vẹn ngắn hạn

Một giao thức an ninh mạng như TLS [RFC2246] hoặc IPSEC [RFC2402] CÓ THỂ được định cấu hình để giúp cho việc so sánh và phân loại các gói tin được truyền qua kết nối mạng.

4.1.4.5 Tính bảo mật dài hạn

Sự mật mã hóa XML là một hoạt động đầu mối W3C/IETF góp phần tích cực trong việc biên soạn một quy định kỹ thuật để mật mã hóa có lựa chọn một (các) tài liệu XML. Dự đoán rằng tiêu chuẩn này được hoàn tất trong vòng năm tới. Các nhóm Transport (truyền tải), Routing (định tuyến) và Packaging (đóng gói) ebXML trong version 1.0 của tiêu chuẩn này đã công nhận công nghệ này như là biện pháp có thể làm được việc cung cấp tính bảo mật có chọn lọc và lâu dài cho các phần tử trong một thông điệp ebXML bao gồm **Header** (tiêu đề) của SOAP.

Tính bảo mật của các phần chứa vùng mang thông tin ebXML CÓ THỂ được cung cấp bởi tính chức năng được tự chủ bằng một MSH. Tính bảo mật của Vùng mang thông tin có thể được tạo ra bằng cách

dùng mật mã hóa XML (mỗi khi có thể) hoặc một số phương pháp mật mã khác (như S/MIME [S/MIME], [S/MIMEV3] hoặc PGP MIME [PGP/MIME]) đã được chấp nhận ở cả hai phía của các nhóm có liên quan. Tiêu chuẩn mật mã hóa XML phải là phương pháp mật mã hóa xác lập mặc định khi mật mã hóa XML đã đạt được trạng thái theo KHUYẾN CÁO của W3C.

CHÚ THÍCH: Khi MSH đòi hỏi cả sự mật mã hóa và ký thì ký trước và sau đó là mật mã hóa.

4.1.4.6 Tính bảo mật ngắn hạn

Một giao thức mạng như TLS [RFC2246] hoặc IPSEC [RFC2402], cung cấp tính bảo mật tạm thời của một thông điệp khi nó được truyền giữa hai nút MSH ebXML cạnh nhau.

4.1.4.7 Quyền dài hạn

Ủy ban dịch vụ kỹ thuật an ninh (TC) OASIS tham gia tích cực trong việc xác định một quy định kỹ thuật để tạo ra sự trao đổi các khả năng an ninh, bao gồm các quyền và sự đòi quyền lợi về tên (Name Assertion and Entitlements) dựa trên ngôn ngữ đánh dấu đòi quyền an ninh (Security Assertion Markup Language [SAML]). Việc sử dụng công nghệ dựa trên quy định kỹ thuật biết trước có thể tạo ra quyền hạn lâu dài cho một thông điệp *ebXML* một khi nó sẵn dùng.

4.1.4.8 Quyền ngắn hạn

Một giao thức an ninh mạng như TLS [RFC2246] hoặc IPSEC [RFC2402] CÓ THỂ được định cấu hình để cung cấp sự xác nhận song phương các chứng chỉ trước khi xác lập một phiên. Việc cung cấp khả năng này cho một MSH ebXML để xác thực nguồn gốc của một sự kết nối và để công nhận nguồn này như một nguồn xác nhận của thông điệp ebXML .

4.1.4.9 Trusted Timestamp (Tem thời gian được chứng thực)

Tại thời điểm của tiêu chuẩn này, các dịch vụ cung cấp các khả năng sẵn dùng. Một khi các dịch vụ này sẵn dùng một cách rộng rãi và có một chuẩn đã được xác định cho sự mở rộng và sử dụng của chúng thì các dịch vụ, các kỹ thuật và các chuẩn này được đánh giá và cân nhắc cho việc sử dụng tiêu chuẩn này cho phiên bản sau.

4.1.5 Xem xét an ninh

Việc thực thi nên được chú thích, hiện tại có một điểm yếu (điểm yếu) ngay cả khi một chữ ký số XML được dùng để bảo vệ tính toàn vẹn cho thông điệp ebXML. Sự quan trọng của điểm yếu tùy thuộc vào môi trường được triển khai áp dụng và truyền tải được dùng để trao đổi các thông điệp ebXML.

Điểm yếu xuất hiện do việc truyền thông điệp ebXML là sự kết hợp cả công nghệ MIME và XML. Bất cứ khi nào có hai hoặc nhiều công nghệ được kết hợp thì luôn nảy sinh thêm vấn đề an ninh. Trong trường hợp này, MIME được dùng như một khung các gói thông điệp, bao gồm phần tử **Envelope** SOAP (đường bao SOAP) và bất cứ phần chứa (container) vùng mang thông tin nào. Các phần tử khác nhau của **Envelope** SOAP (đường bao SOAP) tạo ra sự liên hệ của các chi trả được định danh thông qua các kỹ thuật MIME. Ngoài ra, các nhãn khác nhau được nhân đôi (sao) trong cả phần **Envelope** SOAP

(đường bao SOAP) và khung MIME, ví dụ các loại nội dung trong việc chi trả. Vấn đề là khi nào tất cả các thông tin này được sử dụng và sử dụng như thế nào

Cụ thể, đối với MIME `Content-ID:header` được dùng để chỉ rõ sự duy nhất, định danh nhân cho mỗi vùng mang thông tin. Nhân được dùng trong **Envelope** SOAP (đường bao SOAP) để định danh vùng mang thông tin bất cứ khi nào nó cần. Đối với MIME `Content-Type` (loại nội dung): tiêu đề được dùng để xác định loại nội dung trong vùng mang thông tin, một số loại nội dung có thể bao gồm các tham số thêm vào nhằm nói rõ thêm loại hiện có. Thông tin này sẵn có trong **Envelope** SOAP.

Các tiêu đề của MIME không được bảo vệ, thậm chí cả khi dùng chữ ký số trên cơ sở XML. Mặc dù việc mật mã hóa XML hiện nay không sẵn có để dùng và vì thế không được sử dụng nhưng ứng dụng của nó xây dựng tương tự với chữ ký số XML. Cho tới khi ứng dụng của nó giống với chữ ký số XML thì việc sử dụng của nó sẽ không bảo vệ cho các tiêu đề MIME. Do đó, một thông điệp XML có thể bị nguy hiểm phụ thuộc vào việc thông tin trong các tiêu đề của MIME được xử lý như thế nào khi so sánh với thông tin trong **Envelope** SOAP.

Với `Content-ID:header` MIME là thông tin quyết định. Người xâm nhập có thể dễ dàng cài một phần mềm tấn công phủ nhận dịch vụ bằng cách trộn và làm phù hợp các vùng mang thông tin với `Content-ID:headers`.

Như với hầu hết các sự tấn công phủ nhận dịch vụ, không có sự bảo vệ riêng nào dành cho thông điệp dễ bị tấn công này.

Tuy nhiên, nó cần được nhận ra khi luật vụng đã tính toán cho việc vùng mang thông tin thực tế phải không khớp với luật vụng đã chứa trong SOAP **Envelope** khi chữ ký dạng số được công nhận có giá trị.

Sự có mặt của các loại nội dung trong cả các tiêu đề của MIME và **Envelope** SOAP (đường bao SOAP) là một vấn đề. Các thói quen an ninh thông thường hạn chế việc sao thông tin ở hai nơi. Khi thông tin được sao, thói quen an ninh thông thường yêu cầu thông tin ở hai nơi so sánh với nhau để đảm bảo chúng bằng nhau. Nó phải được coi là sự phá vỡ an ninh khi cả hai tập tin không khớp nhau.

Đối thủ có thể thay đổi các đầu trang của MIME trong khi một thông điệp đang được gửi từ gốc đến đích và điều này có thể không bị phát hiện nếu các dịch vụ an ninh không được xác nhận tính hợp lệ. Trong môi trường truyền tải đồng đẳng (peer-to-peer) thì mối đe dọa này ít nguy hiểm hơn so với môi trường truyền tải đa bước truyền (multi-hop). Tất cả các sự thực thi này là rủi ro khi thông điệp ebXML được ghi lại trong một vùng lưu trữ lâu dài do sự sắp xếp của vùng đó đặt thông điệp vào rủi ro do việc sửa đổi.

Các rủi ro thực tế phụ thuộc vào cách thức một phần mềm thực thi sử dụng sao chép các bộ thông tin. Nếu mọi quá trình xử lý vượt ra ngoài sự phân tách MIME đối với định danh và phân tách phần thân phụ thuộc vào thông tin trong tiêu đề của MIME thì sự thực hiện có cơ gặp nguy hiểm do bị hướng vào các hành động không định hướng trước hoặc không mong muốn. Điều này có thể được khai thác như thế

nào là tốt nhất đối với một lỗi chương trình phổ biến là tràn bộ đệm cho phép, nó phụ thuộc vào sự kiên trì và óc sáng tạo của đối thủ.

Vì vậy, một thực thi có thể làm giảm nguy cơ bằng cách bảo đảm rằng thông tin không được bảo vệ trong các tiêu đề MIME chưa từng được sử dụng được loại ra bởi MIME phân tích mục đích tối thiểu của việc phân tách và định danh các phần thân. Phiên bản của tiêu chuẩn này không khuyến cáo đối với việc có hoặc không một sự thực thi nên so sánh việc sao chép các bộ thông tin và cũng không có hành động nào thực hiện dựa trên kết quả của việc so sánh.

4.2 Môđun điều khiển lỗi

Mục này mô tả một trình điều khiển dịch vụ thông điệp (MSH) thông báo các lỗi mà nó phát hiện trong một thông điệp ebXML tới MSH khác. Môđun điều khiển và thông báo lỗi dịch vụ thông điệp ebXML được coi như một lớp xử lý phía trên lớp xử lý SOAP. Điều này có nghĩa là MSH ebXML về cơ bản là một bộ điều khiển lớp ứng dụng của một thông điệp SOAP từ cấu trúc của bộ xử lý SOAP. Bộ xử lý SOAP có thể tạo ra một thông điệp **Fault** (lỗi) của SOAP nếu nó không thể xử lý thông điệp. Một *MSH gửi (bên gửi MSH)* PHẢI được chuẩn bị để chấp nhận và xử lý các giá trị **Fault** (lỗi) của SOAP.

Phần mềm của MSH ebXML có thể khiến **Fault** (lỗi) của SOAP được tạo ra và gửi lại cho bên gửi SOAP *Thông điệp*. Đối với trường hợp này thông điệp gửi lại PHẢI làm cho thích hợp với chuẩn công nghệ [SOAP] đang xử lý các nguyên tắc đối với các giá trị **Fault** (lỗi) của SOAP.

Một thông điệp đang thông báo một lỗi **highestSeverity** là **Warning** PHẢI KHÔNG được thông báo và gửi lại như **Fault** (lỗi) của SOAP.

4.2.1 Định nghĩa

Để rõ ràng, hai cụm từ được định nghĩa để dùng trong mục này là:

- "Message in error" ("thông điệp lỗi") – Một thông điệp bao gồm hoặc đang gây ra lỗi hoặc đang cảnh báo các lỗi;
- "Message reporting the error" ("thông điệp báo cáo lỗi") – Một thông điệp bao gồm Một phần tử **ErrorList** của ebXML đang mô tả các cảnh báo và (hoặc) các lỗi được tìm thấy trong một thông điệp lỗi (cũng liên quan tới một thông điệp báo cáo lỗi ở một nơi khác trong tài liệu này).

4.2.2 Các loại lỗi

Một MSH cần thông báo các lỗi tới MSH khác. Ví dụ, các lỗi tương ứng với:

- nội dung được hạn định tên miền ebXML của tài liệu *thông điệp* SOAP (xem mục 2.3.1);
- lỗi truyền thông điệp tin cậy (xem mục 4.5.7);
- an ninh (xem mục 4.1).

Trừ quy định ngược lại, tất cả các gì liên quan tới "một lỗi" ("an error") trong phần còn lại của tiêu chuẩn này đưa đến tất cả các loại lỗi được liệt kê ở trên hoặc được xác định ở một nơi khác.

TCVN ISO/TS 15000-2 : 2007

Các lỗi tương ứng với các giao thức truyền thông dữ liệu được phát hiện và sử dụng gián tiếp các kỹ thuật chuẩn được hỗ trợ bởi giao thức truyền thông dữ liệu đó và không sử dụng kỹ thuật thông báo lỗi đã mô tả ở đây.

4.2.3 Phần tử **Errorlist** (danh sách lỗi)

Sự có mặt của phần tử **Errorlist** (danh sách lỗi) trong phần tử **Header** SOAP cho biết thông điệp được định danh bởi **RefToMessageId** (tham chiếu tới id thông điệp) trong phần tử **MessageHeader** (tiêu đề thông điệp) có một lỗi.

Phần tử **Errorlist** gồm có:

- thuộc tính **id** (xem chi tiết phần 2.3.7);
- thuộc tính **version** (phiên bản) (xem chi tiết phần 2.3.8);
- thuộc tính **mustUnderstand** SOAP (phải hiểu) với giá trị của “1” (xem chi tiết phần 2.3.9);
- thuộc tính **highestSeverity** (quy định cao nhất);
- một hoặc nhiều thuộc tính **error** (lỗi).

Nếu không có lỗi nào được thông báo thì phần tử **ErrorList** không được xuất hiện.

4.2.3.1 Thuộc tính **highestSeverity** (tính nghiêm ngặt cao nhất)

Thuộc tính **highestSeverity** (tính nghiêm ngặt cao nhất) bao gồm tính nghiêm ngặt nhất của bất kỳ Một phần tử **Error** (lỗi) nào. Cụ thể là, nếu bất kỳ phần tử **Error** (lỗi) có lỗi nghiêm trọng thì **highestSeverity** (tính nghiêm ngặt cao nhất) PHẢI được thiết lập là **Error**, ngoài ra **highestSeverity** (tính nghiêm ngặt cao nhất) phải được thiết lập là **Warning**.

4.2.3.2 Phần tử **Error** (lỗi)

Một phần tử **Error** (lỗi) gồm:

- thuộc tính **id** (xem chi tiết phần 2.3.7);
- thuộc tính **codeContext** (nội dung mã);
- thuộc tính **errorCode** (mã lỗi);
- thuộc tính **severity** (tính nghiêm ngặt);
- thuộc tính **location** (vị trí).
- phần tử **Description** (mô tả).

4.2.3.2.1 Thuộc tính **id**

Nếu lỗi là một phần của Một phần tử của ebXML, **id** của phần tử đó CÓ THỂ được cung cấp cho quá trình truy vết lỗi.

4.2.3.2.2 Thuộc tính **codeContext** (nội dung mã)

Thuộc tính **codeContext** (nội dung mã) định danh tên miền hoặc giản đồ cho **errorCodes** (các mã lỗi). Nó PHẢI là URI. Giá trị mặc định là **urn:oasis:names:tc:ebxml-msg:service:errors**. Nếu nó không có giá trị mặc định thì sự thực thi tiêu chuẩn này tự nó đã sử dụng các thuộc tính **errorCode**.

Việc sử dụng giá trị thuộc tính **codeContext** khác với mặc định là KHÔNG ĐƯỢC KHUYẾN CÁO. Thêm vào đó, việc thực thi tiêu chuẩn này không nên sử dụng các giá trị thuộc tính **errorCode** (mã lỗi) của chính nó nếu sự có mặt của thuộc tính **errorCode** (mã lỗi) giống như đã xác định trong phần này có ý nghĩa giống nhau hoặc rất giống nhau.

4.2.3.2.3 Thuộc tính **errorCode** (mã lỗi)

Thuộc tính **errorCode** (mã lỗi) YÊU CẦU cho biết bản chất của lỗi trong thông điệp bao gồm lỗi. Các giá trị hợp lệ của thuộc tính **errorCode** (mã lỗi) và sự mô tả nghĩa của mã được đề cập trong phần tới.

4.2.3.2.4 Thuộc tính **severity** (tính nghiêm ngặt)

Thuộc tính yêu cầu **severity** (tính nghiêm ngặt) cho biết tính nguy hiểm của lỗi. Các giá trị hợp lệ là:

- **Warning** – Chỉ ra các thông điệp khác trong hội thoại có thể được tạo ra một cách bình thường bỏ qua các vấn đề;
- **Error** - Chỉ ra có một lỗi không thể khắc phục trong thông điệp và không giúp cho việc xử lý thông điệp phát sinh. Các điều kiện lỗi nên được truyền tới ứng dụng.

4.2.3.2.5 Thuộc tính **location** (vị trí)

Thuộc tính **location** (vị trí) chỉ tới một phần của thông điệp bao gồm lỗi.

Nếu một lỗi tồn tại trong phần tử ebXML và tài liệu chứa được đánh dấu (xem XML [XML]) thì nội dung của thuộc tính **location** (vị trí) PHẢI là Xpointer [Xpointer].

Nếu lỗi tương ứng với *phần chứa vùng mang thông tin* ebXML thì **location** (vị trí) chứa `content-id` của phần MIME đang lỗi, sử dụng giản đồ URI "cid".

4.2.3.2.6 Phần tử **Description** (Mô tả)

Nội dung của phần tử **Description** (mô tả) cung cấp tính chất phần mô tả của lỗi theo ngôn ngữ được xác định bởi thuộc tính **xml:lang**. Sự phân tích XML hoặc phần mềm kiểm tra sự hợp lệ của thông điệp điển hình tạo nên thông điệp đó. Nội dung được xác định bởi nhà cung cấp/người phát triển phần mềm đã tạo ra phần tử **Error** (lỗi) (xem phần 3.1.8).

4.2.3.3 Ví dụ **ErrorList** (danh sách lỗi)

Ví dụ về phần tử **ErrorList** được thể hiện như sau:

```
<eb:ErrorList eb:id="3490sdo", eb:highestSeverity="error" eb:version="2.0"
SOAP:mustUnderstand="1">
  <eb:Error eb:errorCode="SecurityFailure" eb:severity="Error"
eb:location="URI_of_ds:Signature">
```

```
<eb:Description xml:lang="en-US">Validation of signature
failed<eb:Description>
  </eb:Error>
  <eb:Error ...> ... </eb:Error>
</eb:ErrorList>
```

4.2.3.4 Các giá trị errorCode (mã lỗi)

Phần này mô tả các giá trị của thuộc tính **errorCode** (mã lỗi) được sử dụng trong thông điệp thông báo lỗi. Chúng được mô tả trong bảng với 3 tựa đề:

- cột đầu tiên bao gồm giá trị được sử dụng như **errorCode**, ví dụ **SecurityFailure**;
- cột thứ hai bao gồm “Mô tả ngắn” (Short Description) của **errorCode**. Sự tường thuật này không được sử dụng trong nội dung của phần tử **Error**;
- cột thứ 3 bao gồm “Mô tả dài” (Long Description) của **errorCode** (mã lỗi) để cung cấp sự giải thích nghĩa của lỗi và đưa ra sự chỉ dẫn vào lúc **errorCode** (mã lỗi) cá biệt cần được sử dụng.

4.2.3.4.1 Thông báo lỗi trong phần tử ebXML

Danh sách dưới đây chứa các mã lỗi có thể tương ứng với các phần tử ebXML:

Mã lỗi	Mô tả ngắn	Mô tả dài
ValueNotRecognized	Nội dung phần tử hoặc giá trị thuộc tính không được công nhận	Mặc dù tài liệu được đánh dấu đúng ngữ pháp và hợp lệ, phần tử/thuộc tính chứa một giá trị cụ thể không được công nhận vì vậy không được sử dụng bởi <i>dịch vụ thông điệp ebXML</i> .
NotSupported	Phần tử hoặc thuộc tính không hỗ trợ	Mặc dù tài liệu được đánh dấu đúng ngữ pháp và hợp lệ, đơn vị đo là nhất quán với các quy tắc và ràng buộc chứa trong quy định kỹ thuật nhưng cũng không được hỗ trợ bằng <i>dịch vụ thông điệp ebXML</i> để xử lý thông điệp.
Inconsistent	Nội dung phần tử hoặc giá trị thuộc tính không nhất quán với giá trị và phần tử khác	Mặc dù tài liệu được đánh dấu (đúng ngữ pháp) và hợp lệ, theo các nguyên tắc và ràng buộc chứa trong tiêu chuẩn này thì nội dung của Một phần tử hoặc thuộc tính là không nhất quán với nội dung của các phần tử khác hoặc với các thuộc tính của chúng.
OtherXml	Các lỗi khác trong nội dung phần tử và giá trị thuộc tính	Mặc dù tài liệu được đánh dấu đúng ngữ pháp và hợp lệ, nội dung phần tử hoặc giá trị thuộc tính chứa các giá trị không phù hợp với các quy tắc và ràng buộc được chứa trong tiêu chuẩn này và không được đảm bảo bằng các mã khác. Nội dung của phần tử Error nên được sử dụng để chỉ ra bản chất của trục trặc.

4.2.3.4.2 Các lỗi của tài liệu phi XML (non-XML)

Các mã lỗi dưới đây xác định các lỗi không tương ứng với các phần tử ebXML.

Mã lỗi	Mô tả ngắn	Mô tả dài
<i>DeliveryFailure</i>	Lỗi tạo ra thông điệp	Một thông điệp được nhận có thể hoặc chắc chắn không thể được gửi tới đích tiếp theo của nó
<i>TimeToLiveExpired</i>	Kết thúc thông điệp <i>Time To Live</i>	Một thông điệp được nhận đã tới sau thời gian lý thuyết trong phần tử <i>TimeToLive</i> (thời gian làm việc) của phần tử <i>MessageHeader</i> (tiêu đề thông điệp)
<i>SecurityFailure</i>	Lỗi kiểm tra an ninh thông điệp	Hiệu lực của chữ ký hoặc kiểm tra tính xác thực hoặc độ tin cậy của người gửi thông điệp bị lỗi.
<i>MimeProblem</i>	Lỗi giải quyết URI	Nếu thuộc tính <code>xlink:href</code> bao gồm một URI, không có id của nội dung (giản đồ " <code>cid</code> " URI) và URI không được giải quyết, thì nó là một quyết định thực hiện để thông báo lỗi hoặc không.
<i>Unknown</i>	Lỗi không định danh	Cho biết rằng một lỗi đã xảy ra không được bảo trợ của bất kỳ một lỗi nào khác. Nội dung của phần tử <i>Error</i> nên được dùng để chỉ ra bản chất của các lỗi.

4.2.4 Thực hiện việc quản lý và thông báo lỗi

4.2.4.1 Khi tạo ra các thông điệp lỗi

Khi MSH phát hiện ra một lỗi trong thông điệp, nó yêu cầu dứt khoát lỗi được báo cáo tới MSH rằng đã gửi thông điệp lỗi. Điều này có thể xảy ra khi:

- Vị trí báo cáo lỗi (Error Reporting Location) (xem phần 4.2.4.2) mà thông điệp báo cáo lỗi cần được gửi có thể đã được xác định;
- thông điệp lỗi không có phần tử ***ErrorList*** với ***highestSeverity*** thiết lập là ***Error***;

Nếu không thể tìm thấy Vị trí báo cáo lỗi (Error Reporting Location) hoặc thông điệp đang lỗi có phần tử ***ErrorList*** với ***highestSeverity*** thiết lập là ***Error***, nó được quy định như sau:

- lỗi đã được ghi;
- các vấn đề đã được giải quyết bằng biện pháp khác;
- không có hoạt động nào được ghi nhận lại.

4.2.4.2 Xác định vị trí báo cáo lỗi (Error Reporting Location)

Vị trí báo cáo lỗi (Error Reporting Location) là URI trên lý thuyết do người gửi thông điệp lỗi cho biết nơi để gửi thông điệp báo cáo lỗi.

ErrorURI được bao hàm bởi ***CPA*** được định danh bằng ***CPAid*** trên thông điệp, nên được sử dụng. Mặt khác, người nhận có thể giải quyết ***ErrorURI*** bằng cách sử dụng phần tử ***From*** (xuất phát từ) của thông điệp lỗi. Nếu không xảy ra khả năng này, PHẢI không có lỗi nào được báo cáo việc gửi của bên tham gia.

Ngay cả khi thông điệp lỗi không thể được phân tích một cách hoàn chỉnh, các công cụ MSH có thể thử xác định *Vị trí báo cáo lỗi* (Error Reporting Location) bằng các biện pháp khác. Đây là sự thực hiện đầy đủ.

4.2.4.3 Các giá trị của phần tử **Service** (dịch vụ) và phần tử **Action** (hành động)

Phần tử **ErrorList** (danh sách lỗi) có thể được chứa trong **Header** (tiêu đề) của SOAP mà nó là một phần của *thông điệp* được gửi như là kết quả của quá trình của thông điệp ban đầu. Trong trường hợp này, các giá trị cho các phần tử **Service** (dịch vụ) và **Action** (hành động) được thiết lập bởi người thiết kế dịch vụ đó. Phương pháp này không được sử dụng nếu **highestSeverity** là **Error**.

Phần tử **ErrorList** có thể được bao hàm trong *thông điệp* độc lập. Trong trường hợp này, các giá trị của các phần tử **Service** (dịch vụ) và **Action** (hành động) PHẢI được thiết lập như sau:

- phần tử **Service** (dịch vụ) PHẢI được thiết lập là: **urn:oasis:names:tc:ebxml-msg:service**;
- phần tử **Action** (hành động) PHẢI được thiết lập là **MessageError**.

4.3 Môđun **SyncReply** (trả lời đồng bộ)

Điều này có thể cần thiết cho người gửi thông điệp, sử dụng một giao thức liên lạc truyền thông đồng bộ, ví dụ như HTTP, để nhận thông điệp phản hồi kết hợp trên và việc kết nối, thông điệp yêu cầu được phát đi. Trong trường hợp của HTTP, người gửi thông điệp yêu cầu HTTP chứa một thông điệp ebXML cần có thông điệp phản hồi ebXML đã phát tới nó trên và kết nối HTTP.

Nếu có các nút trung gian (hoặc các nút MSH ebXML hoặc các nút SOAP khác) liên quan trong đường dẫn thông điệp, nó là sự cần thiết để cung cấp một vài biện pháp khác bằng cách người gửi thông điệp có thể cho biết nó đang trông đợi sự phản hồi (câu trả lời), vì các nút trung gian có thể duy trì kết nối mở.

Thông điệp mở rộng **SyncReply** SOAP của ebXML được phục vụ cho mục đích này.

4.3.1 Phần tử **SyncReply** (trả lời đồng bộ)

Phần tử **SyncReply** (trả lời đồng bộ) có thể tồn tại giống như các phần tử con trực tiếp của phần tử **Header** SOAP. Nó bao gồm:

- thuộc tính **id** (xem chi tiết phần 2.3.7);
- thuộc tính **version** (phiên bản) (xem chi tiết phần 2.3.8);
- thuộc tính **actor** SOAP với giá trị bắt buộc là "**http://schemas.xmlsoap.org/soap/actor/next**";
- thuộc tính **mustUnderstand** SOAP (phải hiểu) với giá trị là "1" (xem chi tiết phần 2.3.9).

Nếu như hiện tại, phần tử này cho biết nút nhận SOAP hoặc MSH ebXML có liên quan thì thông điệp được nhận nên được duy trì mở trong sự chờ đợi thông điệp phản hồi được quay lại thông qua và một kết nối.

Phần tử này không được sử dụng để ghi đè lên giá trị của **syncReplyMode** (phương thức trả lời đồng bộ) trong CPA. Nếu giá trị của **syncReplyMode** (phương thức trả lời đồng bộ) là **none** và phần tử **SyncReply** đang xuất hiện, thì *MSH nhận* nên đưa ra một lỗi với **errorCode** (mã lỗi) của **Inconsistent** và **severity** của **Error** (xem phần 4.1.5).

Ví dụ của phần tử **SyncReply**:

```
<eb:SyncReply eb:id="3833kkj9" eb:version="2.0" SOAP:mustUnderstand="1"
SOAP:actor="http://schemas.xmlsoap.org/soap/actor/next"/>
```

5 Sự kết nối các phần tử đuôi mở rộng SOAP của ebXML

Phần này mô tả cách các phần tử đuôi mở rộng SOAP của ebXML khác nhau có thể được sử dụng trong kết nối.

5.1.1 Sự tương tác của phần tử **MessageHeader** (Tiêu đề thông điệp)

Phần tử **MessageHeader** (tiêu đề thông điệp) **PHẢI** được xuất hiện trong mọi thông điệp.

5.1.2 Sự tương tác của phần tử **Manifest** (Bảng kê)

Phần tử **Manifest** (bảng kê) **PHẢI** có mặt nếu có bất cứ dữ liệu nào kết hợp với thông điệp không hiện hành trong **Header Container**. Điều này áp dụng chi tiết đối với dữ liệu trong **Payload Container(s)** hoặc một nơi nào khác, ví dụ trên trang web.

5.1.3 Sự tương tác của phần tử **Signature** (Chữ ký)

Một hoặc nhiều phần tử **Signature** (chữ ký) của XML [XMLDSIG] [XMLDSIG] **Signature** (chữ ký) có thể xuất hiện trên bất cứ thông điệp nào.

5.1.4 Sự tương tác của phần tử **ErrorList** (danh sách lỗi)

Nếu thuộc tính **highestSeverity** (tính nghiêm ngặt cao nhất) trên **ErrorList** (danh sách lỗi) được thiết lập là **Warning** thì phần tử này có thể được xuất hiện với bất cứ phần tử nào.

Nếu thuộc tính **highestSeverity** (tính nghiêm ngặt cao nhất) trên **ErrorList** (danh sách lỗi) được thiết lập là **Error**, thì phần tử này có thể xuất hiện với phần tử **Manifest** (bảng kê).

5.1.5 Sự tương tác của phần tử **SyncReply** (trả lời đồng bộ)

Phần tử **SyncReply** (trả lời đồng bộ) có thể tồn tại trên bất cứ thông điệp bên ngoài nào đã gửi sử dụng giao thức liên lạc (kết nối) đồng bộ.

Chương II: Các tính năng bổ sung

6 Môđun thông điệp xác thực

Thông điệp xác thực xác định khả năng tương tác của giao thức giống như hai bộ xử lý dịch vụ thông báo (MSH) có thể tin cậy trao đổi các thông điệp, sử dụng các tin báo đã nhận, thử lại và phát hiện ra bản sao và cơ chế loại trừ dẫn đến bên tham gia To nhận được thông điệp *Once-And-Only-Once* (một và chỉ một). Giao thức linh hoạt cho phép cả việc lưu giữ và chuyển tiếp cho tới thông báo xác thực cuối cùng.

Độ tin cậy đạt được bởi *MSH nhận* phản hồi tới một thông điệp với thông điệp *báo nhận*.

Thông điệp báo nhận là bất cứ thông điệp ebXML nào chứa phần tử **Acknowledgment** (báo nhận). Sự không tương thích để nhận *thông điệp báo nhận* bằng *MSH gửi* CÓ THỂ tạo ra việc thử lại lâu dài cho đến khi *thông điệp báo nhận* được nhận. Hoặc số lần thử lại ấn định đã vượt quá thời gian bên tham gia From (*From Party*) PHẢI thông báo lỗi không tương thích.

Bất cứ khi nào một thông báo tương tự được nhận nhiều hơn một lần, một vài cách thức phát hiện sự đồng nhất và loại trừ PHẢI được đưa ra, thường là xuyên suốt cơ chế lưu trữ lâu dài.

6.1 Kho lưu trữ thường xuyên và hệ thống tương thích (Bộ lưu trữ lâu dài và lỗi hệ thống)

Một MSH hỗ trợ cho thông điệp xác thực PHẢI lưu giữ được các thông điệp gửi và nhận trong *Kho lưu trữ thường xuyên* (*bộ lưu trữ lâu dài*). Trong trường hợp này, *kho lưu trữ thường xuyên* là cách thức lưu trữ dữ liệu mà không bị mất thông tin khi hệ thống bị lỗi hoặc bị gián đoạn.

Đặc tả kỹ thuật này thừa nhận mức độ khác nhau của sự phục hồi có thể được thực hiện phụ thuộc vào kỹ thuật thường sử dụng để lưu trữ dữ liệu. Tuy nhiên, ở một mức độ tối thiểu, kho lưu trữ thường xuyên với các đặc điểm phục hồi của ổ đĩa cứng (hoặc thiết bị tương đương) cần được sử dụng. Nó là yêu cầu cần thiết mà người tiến hành các đặc tính kỹ thuật cần sử dụng kỹ thuật phục hồi đối với lỗi của bất cứ phần cứng đơn lẻ nào hoặc kết cấu phần mềm.

Sau một sự cố hoặc lỗi gián đoạn hệ thống, một MSH PHẢI chắc chắn rằng các thông báo trong kho lưu trữ thường xuyên đã được xử lý nếu như lỗi hệ thống hoặc lỗi gián đoạn không xảy ra sự cố. Cách làm này là một quyết định đúng đắn.

Để hỗ trợ việc lọc các bản sao thông điệp, một *MSH nhận* PHẢI lưu giữ **MessageId** trong kho lưu trữ thường xuyên. Nó cũng đưa ra yêu cầu về các vấn đề cần được đảm bảo trong kho lưu trữ thường xuyên như sau:

- thông điệp hoàn thành, ít nhất cho đến khi thông tin trong thông điệp được đáp ứng đối với ứng dụng hoặc cách thức khác cần để xử lý nó;
- thời gian thông điệp được tiếp nhận, cũng như thông tin có thể được sử dụng để tạo ra sự phản hồi đối với một YÊU CẦU *trạng thái thông điệp* (xem đoạn 51.1);
- thông điệp phản hồi hoàn thành.

6.2 Các phương pháp thực hiện đối với thông điệp xác thực

Việc hỗ trợ cho thông điệp xác thực được thực hiện theo một trong các cách sau:

- sử dụng giao thức thông điệp xác thực ebXML;
- sử dụng các cấu trúc SOAP của ebXML và với các sản phẩm phần mềm thương mại được thiết kế để cung cấp cho các thông điệp xác thực sử dụng giao thức luân phiên.
 - người sử dụng ứng dụng hỗ trợ cho một số đặc điểm, nhất là việc loại trừ trùng lặp hoặc;
 - một vài sự pha trộn của các tùy chọn nói trên trên từng đặc tính cơ bản.

6.3 Các đuôi mở rộng Header (Tiêu đề) SOAP của thông điệp tin cậy

6.3.1 Phần tử **AckRequested** (yêu cầu báo nhận)

Phần tử **AckRequest** (yêu cầu báo nhận) là một tùy chọn mở rộng đối với **Header** (tiêu đề) của SOAP được sử dụng bởi MSH *gửi* để yêu cầu một MSH *nhận*, hoạt động với vai trò của một nhân tố URI được nhận ra trong thuộc tính **actor** (vai diễn) SOAP, trả lại với một *thông điệp báo nhận*.

Phần tử **Ackrequested** (yêu cầu báo nhận) bao gồm:

- thuộc tính **id** (xem chi tiết phần 2.3.7);
- thuộc tính **version** (phiên bản) (xem chi tiết phần 2.3.8);
- thuộc tính **mustUnderstand** SOAP (phải hiểu) với giá trị “1” (xem chi tiết phần 2.3.9);
- thuộc tính **actor** (vai diễn) SOAP;
- thuộc tính **signed** (được ký).

Phần tử này được sử dụng để chỉ ra một MSH *nhận*, hoạt động với vai trò đồng nhất bởi thuộc tính **actor** SOAP, cho dù *Thông điệp báo nhận* có thể xảy ra và nếu như vậy thì thông điệp nên được ký hiệu bằng MSH *nhận*.

Một thông điệp ebXML có thể có 0, 1 hoặc 2 trường hợp của phần tử **AckRequested** (yêu cầu báo nhận). Một nút MSH đơn lẻ chỉ nên đưa vào Một phần tử **AckRequested** (yêu cầu báo nhận). Nếu có hai phần tử **AckRequested** (yêu cầu báo nhận) tồn tại, chúng PHẢI có các giá trị khác nhau cho từng thuộc tính **actor** SOAP. Phần lớn phần tử **AckRequested** (yêu cầu báo nhận) có thể được hướng tới **URI của bên tham gia** với nghĩa To Party MSH (xem phần 2.3.11) với bất kỳ thông điệp nào đưa ra.

6.3.1.1 Thuộc tính **actor** (vai diễn) của SOAP

Phần tử **AckRequested** (yêu cầu báo nhận) PHẢI được hướng tới MSH tiếp theo (Next MSH) hoặc MSH của bên tham gia To (có các lộ trình bước truyền đơn tương đương). Vấn đề này được hoàn thành bao gồm cả SOAP **actor** với giá trị URN và một trong hai ebXML **actor** URNs đã định nghĩa trong phần 2.3.10 và 2.3.11 hoặc bằng cách xoá đi thuộc tính này. Việc mặc định **actor** hướng tới *To Party* MSH.

6.3.1.2 Thuộc tính *signed*

Việc yêu cầu thuộc tính **signed** được dùng bởi *From Party* cho biết có hoặc không có một thông điệp nhận bởi **To Party MSH** nên kết quả trong *To Party* cần điều chỉnh lại ký hiệu *Thông điệp báo nhận*. Bao gồm phần tử **Signature** (chữ ký) [XMLDSIG] giống như mô tả trong phần 4.1. Các giá trị hợp lệ của **signed** là:

- **true** – *Acknowledgment Message (Thông điệp báo nhận)* ký được yêu cầu hoặc;
- **false** – *Acknowledgment Message (Thông điệp báo nhận)* không ký được yêu cầu

Trước khi thiết lập giá trị của thuộc tính **signed** trong **AckRequested**, **MSH** gửi cần kiểm tra khi nào **MSH** nhận hỗ trợ cho *Thông điệp báo nhận* của loại yêu cầu (xem ebCPP).

Khi **MSH** nhận chứa một thông báo với thuộc tính **signed** thiết lập là **true** hoặc **false** thì nó cần PHẢI kiểm chứng xem nó có khả năng hỗ trợ loại *Thông điệp báo nhận* đã yêu cầu hoặc không.

- nếu **MSH** nhận có thể đưa ra loại *Thông điệp báo nhận* yêu cầu thì nó PHẢI gửi trả **MSH** gửi một thông báo chứa 01 phần tử **Acknowledgmen**;
- nếu **MSH** nhận không thể gửi lại *Thông điệp báo nhận* như yêu cầu thì nó PHẢI thông báo lỗi cho **MSH** gửi như **errorCode** (mã lỗi) của **Inconsistent** và **severity** của either **Error** nếu không nhất quán với CPA hoặc **Warning** nếu không hỗ trợ.

6.3.1.3 Mẫu **AckRequested**

Trong ví dụ sau đây, *Thông điệp báo nhận* được yêu cầu gồm một nút **MSH** hoạt động trong vai trò của *To Party* (xem phần 2.3.11) Phần tử **Acknowledgment** được sinh ra PHẢI là cái đích hướng tới nút **MSH** ebXML hoạt động với vai trò của *From Party* và với đường dẫn thông báo đảo ngược (kết thúc tại *acknowledgment*).

```
<eb:AckRequested SOAP:mustUnderstand="1" eb:version="2.0" eb:signed="false"/>
```

6.3.1.4 Sự ảnh hưởng của các phần tử **AckRequested** (yêu cầu báo nhận)

Một phần tử **AckRequested** (yêu cầu báo nhận) PHẢI không được bao hàm một thông điệp với chỉ Một phần tử **Acknowledgment** (báo nhận). Hạn chế này buộc PHẢI chấp nhận để tránh các vòng lặp lâu dài của các *thông điệp báo nhận*. Một thông điệp *báo lỗi* không được bao gồm phần tử **AckRequested** (yêu cầu báo nhận).

6.3.2 Phần tử **Acknowledgment**

Phần tử **Acknowledgment** là một tùy chọn mở rộng để **Header** SOAP sử dụng bởi một Trình quản lý dịch vụ thông điệp (trình quản lý dịch vụ thông điệp) để chỉ ra các Trình quản lý dịch vụ thông điệp (**MSH**) khác mà nó đã nhận được thông báo. Phần tử **RefToMessageld** (id của thông điệp tham chiếu đến) trong phần tử **Acknowledgment** được sử dụng để định danh thông báo được chấp nhận bởi **MessageId** của nó.

Phần tử **Acknowledgmen** bao gồm các phần tử và thuộc tính sau đây:

- thuộc tính **id** (xem chi tiết phần 2.3.7);
- thuộc tính **version** (phiên bản) (xem chi tiết phần 2.3.8);
- thuộc tính SOAP **Acknowledgment** với giá trị “1” (xem chi tiết phần 2.3.9);
- thuộc tính SOAP **actor**;
- thuộc tính **Timestamp**;
- thuộc tính **RefToMessageId**;
- thuộc tính **From**;
- không hoặc nhiều hơn thuộc tính [XMLDSIG] **Reference**.

6.3.2.1 Thuộc tính SOAP actor

Thuộc tính SOAP **actor** của thuộc tính **Acknowledgment** PHẢI có một giá trị tương ứng với thuộc tính **AckRequested** của thông điệp được chấp nhận. Nếu không có thuộc tính **actor** SOAP tồn tại và thuộc tính **Acknowledgment**, kết quả mặc định PHẢI là *To Party MSH* (xem phần 8.1.3).

6.3.2.2 Thuộc tính Timestamp

Phần tử bắt buộc **Timestamp** là giá trị tương trưng cho thời gian mà thông điệp được chấp nhận được nhận bởi *MSH* bằng việc sinh ra các thông báo đã nhận. Nó PHẢI phù hợp với `dateTime` [XMLSchema] và được biểu thị giống như UTC (phần 3.1.6.2).

6.3.2.3 Phần tử RefToMessageId (id của thông điệp tham chiếu đến)

Phần tử bắt buộc **RefToMessageId** bao gồm **MessageId** (thông điệpId) được sinh ra để báo cáo.

6.3.2.4 Phần tử From

Đây là phần tử giống như phần tử **From** (xuất phát từ) ở trong phần tử **MessageHeader** (tiêu đề thông điệp) (xem phần 3.1.1). Tuy nhiên, khi sử dụng trong ngữ cảnh của phần tử **Acknowledgment**, nó chứa từ định danh của **Party** để sinh ra Thông điệp báo nhận.

Nếu phần tử **From** (xuất phát từ) được bỏ qua thì **Party** PHẢI gửi phần tử được định danh bởi phần tử **From** (xuất phát từ) vào phần tử **MessageHeader** (tiêu đề thông điệp).

6.3.2.5 Phần tử Reference [XMLDSIG] (Phần tử tham chiếu)

Thông điệp báo nhận được sử dụng để có thể xác nhận bởi *MSH* bao gồm một hoặc nhiều phần tử **Reference**, từ chữ ký XML [XMLDSIG] tên miền, xuất phát từ *thông điệp được báo nhận* (xem chi tiết phần 6.1.3). Phần tử **Reference** (tham chiếu) PHẢI là tên miền đủ điều kiện đối với tên miền đã nói ở trên và PHẢI phù hợp với việc quy định chữ ký XML [XMLDSIG]. Nếu thông điệp được báo nhận chứa Một phần tử **AckRequested** (yêu cầu báo nhận) với thuộc tính **signed** thiết lập đúng, khi đó danh sách **Reference** [XMLDSIG] là bắt buộc.

Việc xác nhận *Thông điệp báo nhận* cho biết thông điệp gốc đã đạt được mục đích của nó. Việc xác nhận *Thông điệp báo nhận* quy ước xác nhận tính hợp lệ của người gửi thông điệp báo nhận. Tuy nhiên, Thông điệp báo nhận quy ước không cho biết thông điệp nhận được có còn nguyên vẹn hoặc không. Kể

TCVN ISO/TS 15000-2 : 2007

cả một tệp thông điệp gốc trong *Thông điệp báo nhận* cho biết người gửi ban đầu được thừa nhận bởi thông điệp đã được xác định. Tệp chứa trong *Thông điệp báo nhận* có thể được so sánh với tệp của thông điệp gốc.

Nếu các tệp khớp nhau, thông điệp đã nhận được còn nguyên vẹn. Giống như một tệp có sẵn trong thông điệp gốc nếu nó được thể hiện được bao gồm trong phần tử [XMLDSIG] **Signature/Reference** .

Nếu thông điệp gốc được thể hiện, các phần tử [XMLDSIG] **Signature/Reference** của thông điệp gốc PHẢI đồng nhất với các phần tử **Acknowledgment**/[XMLDSIG] **Reference** trong **Acknowledgment Message**. Nếu thông điệp gốc không được thể hiện, phần tử [XMLDSIG] **Reference** PHẢI được bắt nguồn từ thông điệp gốc (xem phần 6.1.3).

Sự xác nhận nói trên của *Thông điệp báo nhận* cuối cùng, *MSH của bên tham gia đến từ (From Party)* có thể thông báo thành công của ứng dụng nảy sinh cho thông điệp tham chiếu. MSH này nên bỏ qua lỗi **Error** kèm theo hoặc *Thông điệp báo nhận* giống với giá trị **RefToMessageId**.

6.3.2.6 Mẫu Acknowledgment

Ví dụ phần tử **Acknowledgment** hướng tới *To Party MSH*:

```
<eb:Acknowledgment SOAP:mustUnderstand="1" eb:version="2.0">
  <eb:Timestamp>2001-03-09T12:22:30</eb:Timestamp>
  <eb:RefToMessageId>323210:e52151ec74:7ffc@xtacy</eb:RefToMessageId>
  <eb:From> <eb:PartyId>uri:www.example.com</eb:PartyId> </eb:From>
</eb:Acknowledgment>
```

6.3.2.7 Việc tự gửi một thông điệp báo nhận

Nếu không có lỗi trong thông điệp nhận được và tự bản thân *Thông điệp báo nhận* gửi đi, không giống như thông điệp chứa dữ liệu vùng mang thông tin thì các phần tử **Service** (dịch vụ) và **Action** (hành động) PHẢI được thiết lập như sau:

- phần tử **Service** (dịch vụ) PHẢI được thiết lập là **urn:oasis:names:tc:ebxml-msg:service**;
- phần tử **Action** (hành động) PHẢI được thiết lập là **Acknowledgment**.

6.3.2.8 Sự tương tác phần tử Acknowledgment

Phần tử **Acknowledgment** CÓ THỂ xuất hiện trên bất cứ thông điệp nào, loại trừ trường hợp chú thích trong phần 6.3.1.4. Một *thông điệp báo nhận* PHẢI không bị gửi trả lại bởi *Thông điệp báo lỗi*.

6.4 Các tham số truyền thông điệp tin cậy

Phần này mô tả các tham số yêu cầu để kiểm soát việc truyền thông điệp tin cậy. Nhiều tham số có thể đạt được từ CPA.

6.4.1 DuplicateElimination (Loại trừ sao chép)

Phần tử **DuplicateElimination** (loại trừ sao chép) PHẢI được sử dụng bởi *MSH của bên tham gia đến từ (From Party)* để cho biết *MSH nhận* loại trừ sự trùng lặp hoặc không (xem phần 6.6 của hoạt động

Việc truyền thông điệp tin cậy). Nếu giá trị của **duplicateElimination** (loại trừ sao chép) trong CPA là **never** thì **duplicateElimination** (loại trừ sao chép) không được xuất hiện.

- nếu **DuplicateElimination** (loại trừ sao chép) xuất hiện – *To Party MSH* PHẢI duy trì thông điệp trong kho lưu trữ cũng như các thông điệp trùng nhau PHẢI được xuất hiện tới *To Party Application* At – Most – Once hoặc;
- nếu **DuplicateElimination** (loại trừ sao chép) không xuất hiện – *To Party MSH* không yêu cầu bảo quản thông điệp trong kho lưu trữ và không yêu cầu kiểm tra các bản sao.

Nếu **DuplicateElimination** (loại trừ sao chép) tồn tại, *To Party MSH* PHẢI thông qua hoạt động của thông điệp xác thực (xem phần 6.6) bởi các bản sao thông điệp được loại bỏ.

Nếu **DuplicateElimination** (loại trừ sao chép) không xuất hiện, *MSH nhận* không yêu cầu kiểm tra bản sao thông điệp sinh ra. Các bản sao thông điệp có thể được sinh ra cho một ứng dụng và kho lưu trữ thông điệp không quy định – Cho dù việc loại trừ bản sao vẫn cho phép.

Nếu *To Party* không thể hỗ trợ cho chức năng được yêu cầu hoặc nếu giá trị của **duplicateElimination** (loại trừ sao chép) trong CPA không thoả mãn các giá trị mặc định của phần tử, *To Party* cần thông báo lỗi cho *From Party* bằng cách sử dụng **errorCode** (mã lỗi) là **Inconsistent** và **Severity** là **Error**.

6.4.2 AckRequested (Yêu cầu báo nhận)

Tham số **AckRequested** được sử dụng bởi *MSH gửi* để yêu cầu *MSH nhận* hoạt động với vai trò của chủ thể URI được định danh trong thuộc tính **actor** SOAP, ngược lại *thông điệp báo nhận* lại chứa phần tử **Acknowledgment**. (xem phần 6.3.1).

6.4.3 Retries (Thử lại)

Thuộc tính **Retries** (thử lại) từ CPA, là một giá trị nguyên chỉ ra số lần nhiều nhất một *MSH gửi* cần cố gắng đọc lại một thông điệp không phản hồi sử dụng giao thức liên lạc giống nhau.

6.4.4 RetryInterval (khoảng thời gian thử lại)

Tham số **RetryInterval** (khoảng thời gian thử lại), từ CPA, là giá trị thời gian, thể hiện thời hiệu phù hợp với kiểu dữ liệu **duration** [XMLSchema]. Giá trị này chỉ rõ thời hạn tối thiểu *MSH gửi* cần chờ đợi giữa **Retries** (thử lại), nếu *Thông điệp báo nhận* không được công nhận hoặc phát hiện ra lỗi liên lạc trong quá trình cố gắng gửi thông điệp. **RetryInterval** (khoảng thời gian thử lại) áp dụng khoảng thời gian giữa việc gửi thông điệp gốc với lần thử lại đầu tiên cũng như trong khoảng thời gian thử lại.

6.4.5 TimeToLive (Thời gian làm việc)

TimeToLive (thời gian làm việc) được định nghĩa trong phần 3.1.6.4

Về phía thông điệp chuyển giao xác thực, **TimeToLive** (thời gian làm việc) PHẢI tuân theo:

$$\text{TimeToLive} > \text{Timestamp} + ((\text{Retries} + 1) * \text{RetryInterval}).$$

Ở đây **Timestamp** đến **MessageData**.

6.4.6 PersistDuration

Tham số **PersistDuration**, từ CPA, là khoảng thời gian tối thiểu, được thể hiện giống như **duration** [XMLSchema], dữ liệu từ một thông điệp xác thực đã gửi được lưu giữ trong *Bộ lưu trữ lâu dài* (Kho lưu trữ thường xuyên) bởi *MSH nhận*.

Nếu tham số **PersistDuration** tương thích khi thông điệp đầu tiên được gửi đi, phát thông điệp không nên gửi lại thông điệp với **MessageId** giống nhau.

Nếu thông điệp không thể gửi thành công trước khi **PersistDuration** hợp quy cách thì *MSH gửi* nên thông báo lỗi (xem phần 6.5.7).

Tham số **TimeStamp** cho thông điệp xác thực đã gửi, dựa vào thông điệp đầu trang và với tham số **PersistDuration** của nó (đã xác định trong PCA) **PHẢI** quan trọng hơn tham số **TimeToLive** của nó dựa vào thông điệp đầu trang.

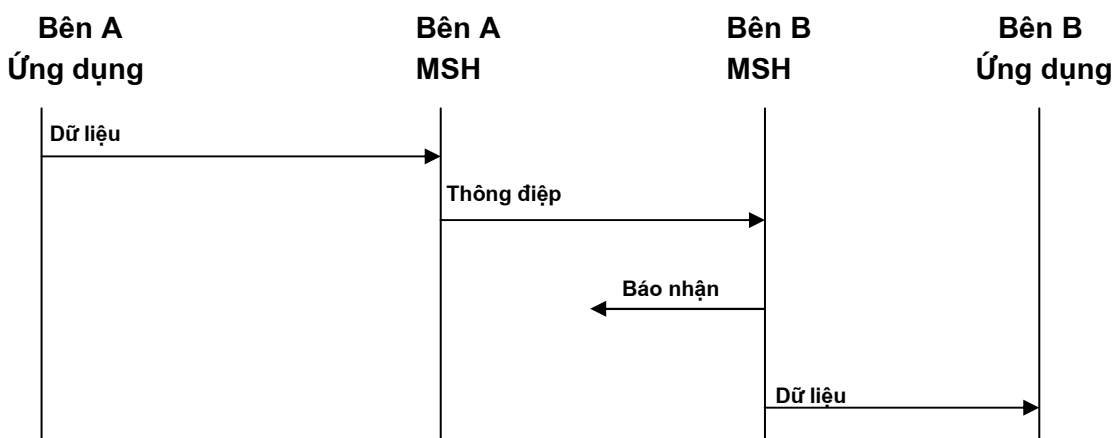
6.4.7 SyncReplyMode (phương thức trả lời đồng bộ)

Tham số **syncReplyMode** từ CPA chỉ được sử dụng nếu giao thức liên lạc dữ liệu là *đồng bộ* (ví dụ HTTP). Nếu giao thức liên lạc *không đồng bộ*, thì giá trị của **syncReplyMode** bị bỏ qua. Nếu thuộc tính **syncReplyMode** không xuất hiện, có nghĩa là nó tương đương với giá trị **none** (không có giá trị). Nếu tham số **syncReplyMode** không **PHẢI** là **none**, phần tử **SyncReply** **PHẢI** hiện hành và MSH **PHẢI** quay trở lại bất kỳ sự phản hồi nào từ ứng dụng hoặc quá trình giao dịch trong vùng mang thông tin của thông điệp phản hồi đồng bộ, theo lý thuyết trong CPA. Các giá trị hợp lệ của **syncReplyMode** là **mshSignalsOnly**, **signalsOnly**, **signalsAndResponse**, **responseOnly** và **none**. Xem thêm phần mô tả **syncReplyMode** trong đặc điểm CPPA [ebCPP].

Nếu giá trị của **syncReplyMode** là **none** và phần tử **SyncReply** đang hiện hữu, thì *MSH nhận* nên thông báo lỗi với **errorCode** (mã lỗi) của **Inconsistent** và **severity** của **Error** (xem phần 4.1.5).

6.5 Giao thức truyền thông điệp tin cậy trong ebXML

Giao thức ebXML Việc truyền thông điệp tin cậy được biểu diễn như hình dạng dưới đây



Hình 6-1 – Cho biết thông điệp đã được nhận

Việc xác nhận *thông điệp báo nhận* cho biết thông điệp được công nhận đã hoàn thành việc tiếp nhận và xử lý hoặc lưu giữ bởi MSH nhận.

Thông điệp báo nhận PHẢI chứa phần tử **Acknowledgment** giống như mô tả ở phần 6.3.1 với phần tử **RefToMessageld** (id của thông điệp tham chiếu đến) chứa giá trị tương đương như phần tử **Messageld** trong thông điệp được công nhận.

6.5.1 Hành vi gửi thông điệp (Sending Message Behavior)

Nếu MSH được đưa ra dữ liệu bằng ứng dụng cần được gửi đi một cách chắc chắn, MSH cần PHẢI làm như sau:

1. tạo một thông điệp từ thành phần được chấp nhận từ ứng dụng;
2. đưa vào một phần tử **AckRequested** như đã định nghĩa trong phần 6.3.1;
3. lưu thông điệp trong kho lưu trữ thường xuyên (xem phần 6.1);
4. gửi thông điệp tới MSH nhận;
5. đợi cho việc quay lại của Thông điệp báo nhận đã được công nhận xác nhận thông điệp riêng biệt này và nếu nó không đến trước khi **RetryInterval** đi qua hoặc nếu bắt gặp lỗi giao thức liên lạc thì tiến hành các hoạt động thích hợp như đã mô tả trong phần 6.5.4.

6.5.2 Hành vi nhận thông điệp (Receiving Message Behavior)

Nếu đây là *Thông điệp báo nhận* như đã được định nghĩa ở phần 6 thì:

1. tìm kiếm thông điệp trong kho lưu trữ thường xuyên với **Messageld** giống như giá trị của **RefToMessageld** trong thông điệp nhận;
2. nếu thông điệp được tìm thấy trong kho lưu trữ thường xuyên thì đánh dấu thông điệp khi phát đi.

Nếu MSH nhận không PHẢI là *To Party* MSH (như định nghĩa tại phần 2.3.10 và 2.3.11) thì xem phần 6.1.3 về cách thức hoạt động của phần tử **AckRequested** (yêu cầu báo nhận).

Nếu phần tử **AckRequested** (yêu cầu báo nhận) xuất hiện (mà không PHẢI là *Thông điệp báo nhận*) thì:

1. Nếu thông điệp là một bản sao (có một **Messageld** được nắm giữ trong kho lưu trữ thường xuyên chứa giá trị giống nhau giống như **Messageld** trong thông điệp nhận) tạo ra Phần tử Acknowledgment (xem phần 6.5.3). Tiếp theo, quy trình trong phần 6.5.5 về việc gửi lại *Thông điệp báo nhận* bị mất. MSH nhận PHẢI không được chuyển giao thông báo tới giao diện ứng dụng.

CHÚ THÍCH: Việc kiểm tra các bản sao chỉ được thực hiện khi có mặt **DuplicateElimination**.

2. nếu thông điệp không PHẢI là bản sao hoặc (không có **Messageld** được giữ lại trong kho lưu trữ thường xuyên tương ứng với **Messageld** trong thông điệp nhận) thì:

- a. nếu là phần tử **DuplicateElimination**, lưu lại **MessageId** của thông điệp nhận trong kho lưu trữ thường xuyên. Như một quyết định đầy đủ và cần thiết, toàn bộ thông điệp có thể được lưu trữ;
- b. tạo ra một *thông điệp báo nhận* trong sự phản hồi (điều này có thể giống như một phần của thông điệp khác). MSH nhận không được gửi thông điệp *báo nhận* cho đến khi thông điệp đã được lưu giữ an toàn trong kho lưu trữ thường xuyên hoặc gửi tới giao diện ứng dụng. Việc phát đi của *Thông điệp báo nhận* tạo nên một trách nhiệm bởi MSH nhận để chuyển giao thông điệp tới giao diện hoặc hướng tới một đường dẫn tiếp theo trong đường dẫn thông điệp một cách thích hợp.

Nếu không có phần tử **AckRequested** (yêu cầu báo nhận) thì thực hiện như sau:

1. nếu có phần tử **DuplicateElimination** (loại trừ sao chép) và thông điệp là một bản sao thì không cần PHẢI thực hiện động tác nào;
2. mặt khác, chuyển giao thông điệp tới giao diện ứng dụng.

Nếu nút MSH nhận đang mở khi vật trung gian tiến về phía đường dẫn thông điệp, thì nó có thể sử dụng cách thức hoạt động lưu trữ và xúc tiến. Tuy nhiên, nó PHẢI không để lọt ra các bản sao thông điệp từ việc xử lý thông thường ở các nút đó.

Nếu *Thông điệp báo nhận* nhận được một cách bất ngờ thì nó PHẢI được bỏ qua. Không có lỗi nào cần PHẢI được bỏ đi.

6.5.3 Tạo thông điệp báo nhận

Một *thông điệp báo nhận* PHẢI được sinh ra mỗi khi thông điệp được nhận với phần tử **AckRequested** (yêu cầu báo nhận) có URI của người tham gia SOAP hướng tới nút MSH nhận.

Tối thiểu nó PHẢI bao gồm phần tử **Acknowledgment** cùng với phần tử **RefToMessageId** (id của thông điệp tham chiếu đến) có cùng giá trị như phần tử **MessageId** trong thông điệp được thừa nhận. Thông điệp này PHẢI được đặt trong kho lưu trữ thường xuyên với phần tử **PersistDuration** giống như thông điệp gốc.

Phần tử **Acknowledgment** có thể được gửi và một lúc giống như sự phản hồi đối với thông điệp nhận. Trong trường hợp này, các giá trị cho phần tử **MessageHeader** (tiêu đề thông điệp) của *Thông điệp báo nhận* đã được xác định bởi **Service** và **Action** kết hợp với sự phản hồi giao dịch.

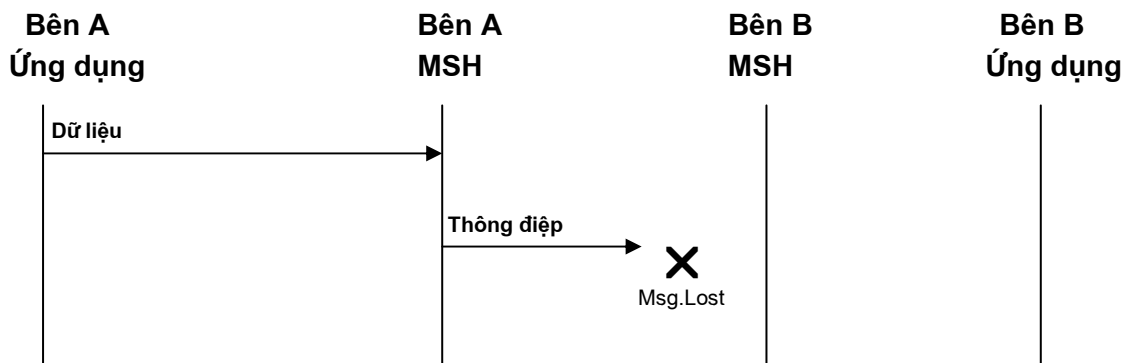
Nếu tự *Thông điệp báo nhận* được gửi đi thì giá trị của các phần tử **Thông điệpHeade** PHẢI được thiết lập như sau:

- phần tử **Service** (dịch vụ) PHẢI được thiết lập với: **urn:oasis:names:tc:ebxml-msg:service;**
- phần tử **Action** (hành động) PHẢI được thiết lập là **Acknowledgment;**
- phần tử **From** (xuất phát từ) có thể được đi và với phần tử **To** (hướng đến) trích từ thông điệp nhận và tất cả các phần tử chịu ảnh hưởng từ phần tử **To** (hướng đến) nên được bao gồm trong phần tử **From** (xuất phát từ) này;

- phần tử **To** (hướng đến) có thể được đi và với phần tử **From** (xuất phát từ) trích từ thông điệp nhận và tất cả các phần tử chịu ảnh hưởng từ phần tử **From** (xuất phát từ) nên được bao gồm trong phần tử **To** (hướng đến) này;
- phần tử **RefToMessageld** (id của thông điệp tham chiếu đến) PHẢI được thiết lập là **Messageld** của thông điệp nhận.

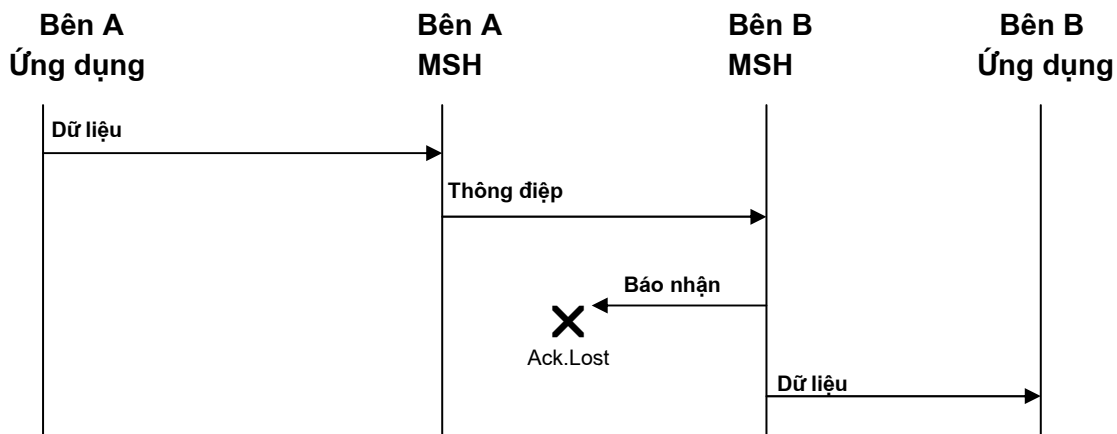
6.5.4 Gửi lại thông điệp đã mất

Phần này mô tả cách thức yêu cầu đối với phía người gửi và nhận thông điệp để quản lý thông điệp bị mất. Một thông điệp được coi là bị “mất” khi *MSH gửi* không nhận được thông báo xác nhận đối với thông điệp. Ví dụ, một thông điệp có thể coi là bị mất là:



Hình 6-2 – Thông điệp không được phát đi

Cũng có thể *Thông điệp báo nhận* bị mất, ví dụ:



Hình 6-3 – Mất thông điệp báo nhận

CHÚ THÍCH: Phần tử **Acknowledgment** không bao giờ báo đã nhận.

Các quy tắc áp dụng đối với việc không xác nhận quyền được biết trước của **Acknowledgment** đối với việc mất thông điệp ứng dụng hoặc *Thông điệp báo nhận* như sau:

- MSH gửi PHẢI gửi lại thông điệp gốc nếu *Thông điệp báo nhận* đã được yêu cầu nhưng chưa được thừa nhận và các vấn đề sau là đúng;

TCVN ISO/TS 15000-2 : 2007

- ít nhất thời gian lý thuyết trong tham số **RetryInterval** đã qua khi thông điệp được gửi đi lần cuối cùng;
 - thông điệp được gửi lại ít hơn số lần trên danh nghĩa trong tham số **Retries**.
- nếu MSH gửi không nhận Thông điệp báo nhận sau số lần thử lại lớn nhất, MSH gửi PHẢI thông báo cho ứng dụng và/hoặc hệ thống quản trị chức năng của tình trạng không thích hợp để nhận *Thông điệp báo nhận* (xem phần 4.2.3.2.4) liên quan đến việc xử lý lỗi);
- nếu MSH gửi phát hiện ra lỗi giao thức kết nối, MSH gửi PHẢI gửi lại thông báo sử dụng thuật toán giống nhau, nếu như nó không nhận được *Thông điệp báo nhận* phần tử.

6.5.5 Gửi lại thông điệp báo nhận

Nếu MSH nhận được một thông điệp mà nó nhận ra là bản sao thì nó nên gửi lại *thông điệp báo nhận* nếu thông điệp được lưu trữ trong kho lưu trữ thường xuyên. Trong trường hợp này, cần thực hiện như sau:

Để ý sự phản hồi đầu tiên của thông điệp nhận trong kho lưu trữ thường xuyên (nó bao gồm Một phần tử **RefToMessageId** (id của thông điệp tham chiếu đến) khớp với phần tử **MessageId** của thông điệp nhận).

Nếu một thông điệp phản hồi được tìm thấy trong kho lưu trữ thường xuyên thì gửi trả lại thông điệp tồn tại cho MSH cái mà đã cho gửi thông điệp nhận. Nếu không có thông điệp phản hồi nào được tìm thấy trong kho lưu trữ thì:

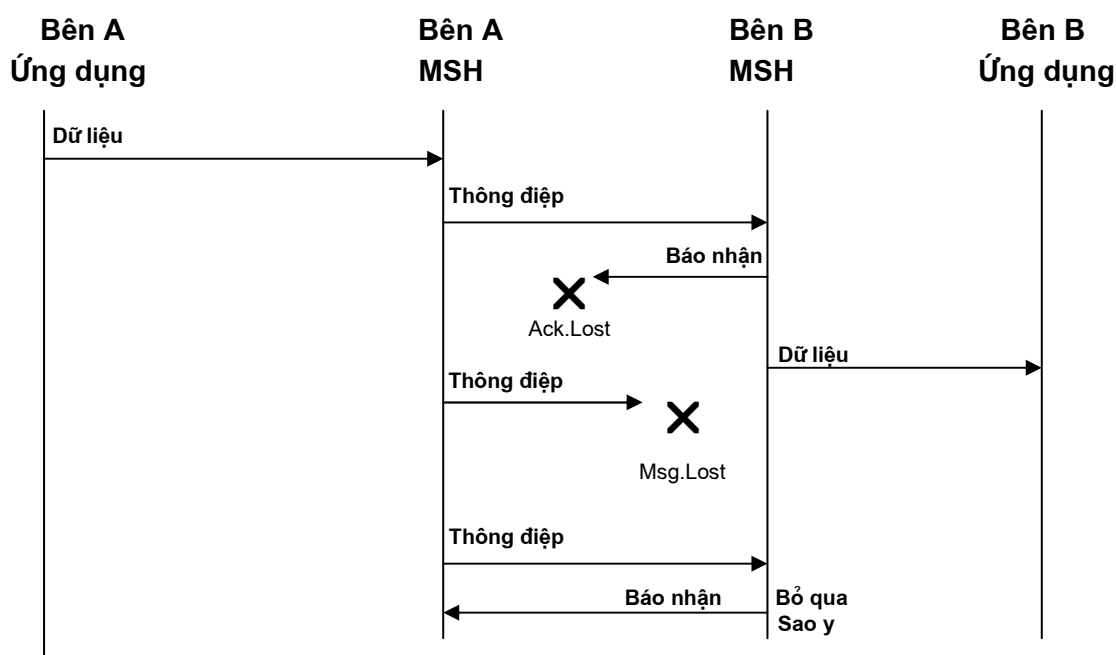
(1) Nếu **syncReplyMode** không được thiết lập tới **none** và nếu CPA cho biết một ứng dụng phản hồi được bao gồm trong đó, thì nó PHẢI là trường hợp mà ứng dụng không kết thúc xử lý quá trình sao chép sớm hơn của và một thông điệp. Bởi vậy, hãy chờ đợi sự phản hồi lại từ ứng dụng và sau đó trả lại sự phản hồi đó đồng bộ với và một kết nối mà đã được sử dụng để truyền lại.

(2) Mặt khác sinh ra một *thông điệp báo nhận*.

6.5.6 Quản lý bản sao thông điệp

Theo nội dung của tiêu chuẩn này:

- “thông điệp đồng dạng” - Thông điệp bao gồm ebXML **Header** SOAP, **Body** và phần chứa đựng vùng mang thông tin ebXML giống với thông điệp đã gửi sớm;
- “bản sao thông điệp” - Thông điệp và phần tử **MessageId** giống với thông điệp đã nhận trước đó;
- “thông điệp phản hồi đầu tiên” - Thông điệp với **Timestamp** đầu tiên trong phần tử **MessageData** (dữ liệu thông điệp) có và **RefToMessageId** giống với bản sao thông điệp.



Hình 6.4 – Việc gửi lại các thông điệp không báo nhận

Sơ đồ trên cho thấy hoạt động của việc *Gửi* và *Nhận* MSH cho các thông điệp đã gửi với phần tử **AckRequested** (yêu cầu báo nhận) và phần tử **DuplicateElimination**. Cụ thể:

1. người gửi thông điệp (ví dụ MSH của bên A) **PHẢI** gửi lại “các thông điệp đồng dạng” nếu không nhận được *Thông điệp báo nhận*;
2. khi người nhận thông điệp (ví dụ MSH của bên B) nhận được “bản sao thông điệp” thì nó **PHẢI** gửi lại người gửi (MSH của bên A) *Thông điệp báo nhận* giống hệt với thông điệp phản hồi đầu tiên cho người gửi (MSH của bên A);
3. người nhận thông điệp (MSH của bên B) không được đẩy tới thông điệp lần thứ 2 cho ứng dụng/quá trình.

6.5.7 Sự truyền phát thông điệp lỗi (Failed Thông điệp Delivery)

Nếu một thông điệp gửi đi với phần tử **Ack Requested** không thể được truyền tải, MSH hoặc quá trình quản lý thông điệp (như một lộ trình trung gian) **PHẢI** gửi thông báo lỗi phát sinh tới *From Party*. Thông điệp thông báo lỗi phát sinh là một thông điệp lỗi (*Thông điệp báo lỗi*) với **errorCode** (mã lỗi) of **DeliveryFailure** và **severity** của:

- **Error** nếu như nhóm phát hiện ra vấn đề không thể truyền đi thông điệp (ví dụ Sự truyền đạt thông báo không có hiệu lực);
- **Warning** nếu như thông điệp đã được truyền đi nhưng Phần tử **Acknowledgment** chưa nhận được. Điều này có nghĩa là, có thể thông điệp chưa truyền đi được.

Có thể nó là một thông điệp báo lỗi với phần tử **Error** có **ErrorCode** (mã lỗi) thiết lập theo **DeliveryFailure** không thể phát đi thành công vì một vài lý do. Nếu điều này xảy ra thì *From Party*, cái

đích cuối và của Thông điệp báo lỗi, PHẢI được hiểu vấn đề theo các ý nghĩa khác. Cách mà việc này được thực hiện bên ngoài phạm vi của thuyết minh này.

CHÚ THÍCH: Nếu MSH của bên tham gia đến từ (From Party) nhận được *Thông điệp báo nhận* từ *To Party* MSH thì nên bỏ qua tất cả các **DeliveryFailure** hoặc *Thông điệp báo nhận* khác.

6.6 Các kết hợp của việc truyền thông điệp tin cậy

	<i>Duplicate-Eliminations</i>	<i>AckRequested ToPartyMSH</i>	<i>AckRequested NextMSH</i>	Chú thích
1	Y	Y	Y	Once-And-Only-Once: Thông báo xác thực ở end - to - end và At - Least - Once tới bộ phận trung gian. Bộ phận trung gian và To Party có thể tạo ra thông báo lỗi (Lỗi truyền Notifications) nếu chúng không thể truyền đi.
2	Y	Y	N	Once-And-Only-Once: Thông điệp xác thực ở mức End - To - End chỉ dựa trên sự truyền lại của end - to - end ở trên.
3	Y	N	Y	At-Least-Once: Thông điệp xác thực ở mức độ trung gian Once-And-Only-Once end-to-end nếu tất cả mức trung gian được xác định. Không có thông báo End - to - End.
4	Y	N	N	At-5Most-Once: Việc loại trừ trùng lặp chỉ ở <i>To Party</i> , không thử lại ở mức trung gian hoặc mức cuối và (the End)
5	N	N	Y	At-Least-Once Thông điệp xác thực với các bản sao có thể có ở mức trung gian hoặc <i>To Party</i>
6	N	Y	N	At-Least-Once: Các bản sao thông điệp xác thực có khả năng ở mức trung gian hoặc <i>To Party</i> .
7	N	N	Y	At-Least-Once Thông điệp xác thực tới mức trung gian và mức cuối cùng. Không có thông báo End - To - End.
8	N	N	N	Sự cố gắng tốt nhất

* Việc loại trừ trùng lặp chỉ thực hiện ở *To Party* MSH, không thực hiện ở mức Trung gian (Intermediate Level).

7 Dịch vụ báo cáo trạng thái thông điệp

Dịch vụ yêu cầu trạng thái thông điệp bao gồm các thành phần sau đây:

- A Thông điệp yêu cầu báo cáo trạng thái thông điệp bao gồm các chi tiết liên quan một thông điệp được gửi trước đó tới Trình quản lý dịch vụ thông điệp (MSH);

- trình quản lý dịch vụ thông điệp nhận các phản hồi yêu cầu cùng với một thông điệp phản hồi trạng thái thông điệp.

Trình quản lý dịch vụ thông điệp NÊN phản hồi tới các yêu cầu trạng thái thông điệp đối với các thông điệp đã được gửi tin cậy và **MessageId** trong **RefToMessageId** tồn tại trong bộ lưu trữ lâu dài (xem phần 6.1).

Trình quản lý dịch vụ thông điệp CÓ THỂ phản hồi tới Các yêu cầu trạng thái thông điệp đối với các thông điệp không được gửi một cách tin cậy.

Dịch vụ thông điệp KHÔNG NÊN sử dụng Dịch vụ yêu cầu trạng thái thông điệp để thực thi Việc truyền thông điệp tin cậy.

Nếu *MSH nhận* không hỗ trợ dịch vụ được yêu cầu, nó NÊN trả lại một thông điệp *báo lỗi* cùng với một **errorCode** (mã lỗi) của **NotSupported** và một thuộc tính **highestSeverity** đặt là **Error**. Mỗi dịch vụ được mô tả như sau.

7.1 Các thông điệp báo cáo trạng thái thông điệp

7.1.1 Thông điệp yêu cầu báo cáo trạng thái thông điệp

Thông điệp yêu cầu báo cáo trạng thái thông điệp bao gồm một thông điệp ebXML with no *phần chứa vùng mang thông tin ebXML* và các phần tử sau đây:

- phần tử **MessageHeader** (tiêu đề thông điệp) bao gồm:
 - phần tử **From** (xuất phát từ) xác định *Party* tạo Thông điệp yêu cầu báo cáo trạng thái thông điệp;
 - phần tử **To** (hướng đến) xác định một *Party* nên nhận thông điệp đó;
 - phần tử **Service** (dịch vụ) bao gồm: **urn:oasis:names:tc:ebxml-msg:service;**
 - phần tử **Action** (hành động) bao gồm **StatusRequest;**
 - phần tử **MessageData** (dữ liệu thông điệp).
- phần tử **StatusRequest** bao gồm:
 - phần tử **RefToMessageId** (id của thông điệp tham chiếu đến) trong Phần tử StatusRequest bao gồm MessageId của thông điệp mà trạng thái của nó đang được truy vấn.
- phần tử **Signature** (chữ ký) [XMLDSIG] (xem phần 4.1).

Thông điệp sau đó được gửi tới *To Party*.

7.1.2 Thông điệp phản hồi trạng thái thông điệp

Ngay khi *To Party* nhận Thông điệp yêu cầu báo cáo trạng thái thông điệp, chúng NÊN tạo một thông điệp phản hồi trạng thái thông điệp không gồm *phần chứa vùng mang thông tin ebXML* bao gồm các phần tử sau đây:

- phần tử **MessageHeader** (tiêu đề thông điệp) bao gồm:
 - phần tử **From** (xuất phát từ) để xác định người gửi của thông điệp phản hồi trạng thái của thông điệp;
 - phần tử **To** (hướng đến) đặt giá trị của phần tử **From** (xuất phát từ) trong Thông điệp yêu cầu báo cáo trạng thái thông điệp;
 - phần tử **Service** (dịch vụ) bao gồm **urn:oasis:names:tc:ebxml-msg:service;**
 - phần tử **Action** (hành động) bao gồm **StatusResponse;**
 - phần tử **MessageData** (dữ liệu thông điệp) bao gồm:
 - **RefToMessageId** để xác định thông điệp yêu cầu báo cáo trạng thái thông điệp.
- phần tử **StatusResponse** (xem phần 52.3) 1815);
- Một phần tử **Signature** (chữ ký) [XMLDSIG] (xem phần 4.1) .

Thông điệp sau đó được gửi tới *To Party*.

7.1.3 Các xem xét về an ninh

Các bên tham gia nhận một thông điệp yêu cầu báo cáo trạng thái thông điệp NÊN luôn luôn phản hồi tới thông điệp đó. Tuy nhiên, chúng CÓ THỂ lược bỏ thông điệp thay cho việc phản hồi cùng với **MessageStatus** đặt là **Unauthorized** nếu chúng xem người gửi của thông điệp là trái phép. Quá trình quyết định trong giai đoạn này của hành động là độc lập thực thi.

7.2 Phần tử StatusRequest (yêu cầu trạng thái)

Phần tử **StatusRequest** (yêu cầu trạng thái) TỰY CHỌN là Một phần tử con trung gian của phần tử **Body** (thân) SOAP và được sử dụng để xác định một thông điệp trước đó mà trạng thái của nó được yêu cầu (xem phần 53.5).

Phần tử **StatusRequest** (yêu cầu trạng thái) bao gồm các phần tử sau:

- một thuộc tính **id** (xem phần 2.3.7);
- một thuộc tính **version** (phiên bản) (xem phần 2.3.8);
- Một phần tử **RefToMessageId** (id của thông điệp tham chiếu đến).

7.2.1 Phần tử **RefToMessageId** (id của thông điệp tham chiếu đến)

Một phần tử **RefToMessageId** (id của thông điệp tham chiếu đến) YÊU CẦU bao gồm **MessageId** (id của thông điệp) của thông điệp mà trạng thái của nó được yêu cầu.

7.2.2 Ví dụ về **StatusRequest** (yêu cầu trạng thái)

Một ví dụ về phần tử **StatusRequest** cho trước dưới đây:

```
<eb:StatusRequest eb:version="2.0" >
  <eb:RefToMessageId>323210:e52151ec74:-7ffc@xtacy</eb:RefToMessageId>
</eb:StatusRequest>
```

7.2.3 Sự tương tác của phần tử **StatusRequest** (yêu cầu trạng thái)

Một phần tử **StatusRequest** (yêu cầu trạng thái) KHÔNG PHẢI có mặt cùng với các phần tử sau đây:

- Một phần tử **Manifest** (bảng kê);
- Một phần tử **StatusResponse** (phản hồi trạng thái);
- Một phần tử **ErrorList** (danh sách lỗi).

7.3 Phần tử **StatusResponse** (phản hồi trạng thái)

Phần tử **StatusResponse** (phản hồi trạng thái) tùy chọn là một con trỏ trung gian của một thân SOAP và được sử dụng bởi một MSH để mô tả trạng thái xử lý của một thông điệp.

Phần tử **StatusResponse** (phản hồi trạng thái) bao gồm các thuộc tính và phần tử sau đây:

- Một thuộc tính **id** (xem phần 2.3.7);
- Một thuộc tính **version** (phiên bản) (xem phần 2.3.8);
- Một phần tử **RefToMessageId** (id của thông điệp tham chiếu đến);
- Một phần tử **Timestamp** (tem thời gian);
- Một thuộc tính **MessageStatus** (trạng thái thông điệp).

7.3.1 Phần tử **RefToMessageId** (id của thông điệp tham chiếu đến)

Một phần tử **RefToMessageId** (id của thông điệp tham chiếu đến) YÊU CẦU bao gồm **MessageId** của thông điệp mà trạng thái của nó đang được báo cáo. phần tử **RefToMessageId** (id của thông điệp tham chiếu đến) **con của** phần tử **MessageData** (dữ liệu thông điệp) của một thông điệp bao gồm Một phần tử **StatusResponse** PHẢI có **MessageId** của thông điệp bao gồm phần tử **StatusRequest** cho phần tử **StatusResponse** áp dụng. Phần tử **RefToMessageId** (id của thông điệp tham chiếu đến) con của phần tử **StatusRequest** hoặc **StatusResponse** PHẢI bao gồm **MessageId** của thông điệp mà trạng thái của nó đang được truy vấn.

7.3.2 Phần tử **Timestamp** (Tem thời gian)

phần tử **Timestamp** (tem thời gian) bao gồm thông điệp về thời gian, mà trạng thái của nó đang được báo cáo, đã nhận (phần 3.1.6.2.). Điều này Có thể được lược bỏ nếu thông điệp, mà trạng thái của nó đang được báo cáo, là **NotRecognized** (không công nhận) hoặc trạng thái yêu cầu là **Unauthorized**.

7.3.3 Thuộc tính **MessageStatus** (Trạng thái thông điệp)

Thuộc tính **MessageStatus** YÊU CẦU xác định trạng thái của thông điệp được xác định bởi phần tử **RefToMessageId**. Nó PHẢI được đặt là một trong các giá trị sau đây:

- **Unauthorized** – Yêu cầu trạng thái thông điệp không được cấp quyền hoặc không được chấp nhận
- **NotRecognized** (không công nhận) – thông điệp được xác định bởi phần tử **RefToMessageId** (id của thông điệp tham chiếu đến) trong phần tử **StatusResponse** không được công nhận
- **Received** – Thông điệp được xác định bởi phần tử **RefToMessageId** (id của thông điệp tham chiếu đến) trong phần tử **StatusResponse** đã nhận được bởi MSH
- **Processed** – Thông điệp được xác định bởi phần tử **RefToMessageId** (id của thông điệp tham chiếu đến) trong phần tử **StatusResponse** đã được xử lý bởi MSH
- **Forwarded** – Thông điệp được xác định bởi phần tử **RefToMessageId** (id của thông điệp tham chiếu đến) trong phần tử **StatusResponse** đã được chuyển tiếp bởi MSH tới một MSH khác

CHÚ THÍCH: Nếu một YÊU CẦU trạng thái thông điệp được gửi sau thời gian đã trôi qua được chỉ ra bởi **PersistDuration** đã trải qua khi thông điệp đang được truy vấn được gửi, phản hồi trạng thái thông điệp có thể chỉ ra **MessageId** là **NotRecognized** (không công nhận) – **MessageId** không được lưu trữ trong bộ nhớ lâu dài nữa.

7.3.4 Ví dụ về **StatusResponse** (phản hồi trạng thái)

Một ví dụ về phần tử **StatusResponse** cho trước dưới đây:

```
<eb:StatusResponse eb:version="2.0" eb:MessageStatus="Received">  
  <eb:RefToMessageId>323210:e52151ec74:-7ffc@xtacy</eb:RefToMessageId>  
  <eb:Timestamp>2001-03-09T12:22:30</eb:Timestamp>  
</eb:StatusResponse>
```

7.3.5 Phần tử **StatusResponse Interaction**

Phần tử này KHÔNG PHẢI có mặt cùng với các phần tử sau đây:

- Một phần tử **Manifest** (bảng kê);
- Một phần tử **StatusRequest** (yêu cầu trạng thái);
- Một phần tử **ErrorList** (danh sách mã) cùng với một thuộc tính **highestSeverity** (tính nghiêm ngặt cao nhất) đặt là **Error** (lỗi).

8 Dịch vụ ping của trình quản lý dịch vụ thông điệp

Dịch vụ ping của trình quản lý dịch vụ thông điệp TỰY CHỌN cho phép một MSH xác định một MSH khác đang hoạt động. nó bao gồm:

- một MSH gửi thông điệp ping của trình quản lý dịch vụ tới một MSH và;
- một MSH khác, nhận tín hiệu Ping, phản hồi cùng với thông điệp Pong của trình quản lý dịch vụ thông điệp.

Nếu một *MSH nhận* không hỗ trợ dịch vụ được yêu cầu, nó **NÊN** trả lại một thông điệp *báo lỗi* cùng với một **errorCode** (mã lỗi) là **NotSupported** và một thuộc tính **highestSeverity** đặt là **Error**.

8.1 Thông điệp ping của trình quản lý dịch vụ

Thông điệp Ping (MSH Ping) của trình quản lý dịch vụ thông điệp bao gồm một thông điệp ebXML không bao gồm *phần chứa vùng mang thông tin ebXML* và các phần tử sau đây:

- Một phần tử **MessageHeader** (tiêu đề thông điệp) bao gồm following;
- Một phần tử **From** (xuất phát từ) xác định bên tạo thông điệp Ping MSH;
- Một phần tử **To** (hướng đến) xác định bên gửi thông điệp Ping MSH;
- Một phần tử **CPAId** (id của CPA);
- Một phần tử **ConversationId** (id của hội thoại);
- Một phần tử **Service** (dịch vụ) bao gồm: **urn:oasis:names:tc:ebxml-msg:service;**
- Một phần tử **Action** (hành động) bao gồm **Ping;**
- Một phần tử **MessageData** (dữ liệu thông điệp);
- Một phần tử **Signature** (chữ ký) [XMLDSIG] (xem phần 4.1).

Thông điệp sau đó được gửi tới *To Party*.

Một ví dụ về Ping:

...các tiêu đề truyền

```

SOAPAction: "ebXML"
Content-type: multipart/related; boundary="ebXMLBoundary"

--ebXMLBoundary
Content-Type: text/xml
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
    http://www.oasis-open.org/committees/ebxml-
msg/schema/envelope.xsd">
  <SOAP:Header xmlns:eb="http://www.oasis-open.org/committees/ebxml-
msg/schema/msg-header-2_0.xsd"
    xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-
msg/schema/msg-header-2_0.xsd
      http://www.oasis-open.org/committees/ebxml-
msg/schema/msg-header-2_0.xsd">
    <eb:MessageHeader version="2.0" SOAP:mustUnderstand="1"
      xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-
2_0.xsd"

```

```

    xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-
msg/schema/msg-header-2_0.xsd
                        http://www.oasis-open.org/committees/ebxml-
msg/schema/msg-header-2_0.xsd">
    <eb:From> <eb:PartyId>urn:duns:123456789</eb:PartyId> </eb:From>
    <eb:To> <eb:PartyId>urn:duns:912345678</eb:PartyId> </eb:To>
    <eb:CPAId>20001209-133003-28572</eb:CPAId>
    <eb:ConversationId>20010215-111213-28572</eb:ConversationId>
    <eb:Service>urn:oasis:names:tc:ebxml-msg:service</eb:Service>
    <eb:Action>Ping</eb:Action>
    <eb:MessageData>
        <eb:MessageId>20010215-111212-28572@example.com</eb:MessageId>
        <eb:Timestamp>2001-02-15T11:12:12</eb:Timestamp>
    </eb:MessageData>
    </eb:MessageHeader>
</SOAP:Header>
<SOAP:Body/>
</SOAP:Envelope>

--ebXMLBoundary--

```

CHÚ THÍCH: Ví dụ trên chỉ ra một cấu trúc MIME quan hệ đa phần với chỉ một bodypart

8.2 Thông điệp Pong của trình quản lý dịch vụ thông điệp

Ngay khi *To Party* nhận thông điệp Ping MSH, chúng CÓ THỂ tạo thông điệp Pong (MSH Pong) của Trình quản lý dịch vụ thông điệp bao gồm một thông điệp ebXML không bao gồm phần chứa vùng mang thông tin ebXML và các phần tử sau đây:

- Một phần tử **MessageHeader** (tiêu đề thông điệp) bao gồm các phần tử sau đây:
 - Một phần tử **From** (xuất phát từ) xác định người tạo Thông điệp Pong MSH;
 - Một phần tử **To** (hướng đến) xác định bên đã tạo ra thông điệp Ping MSH;
 - Một phần tử **CPAId** (id của CPA);
 - Một phần tử **ConversationId** (id của hội thoại);
 - Một phần tử **Service** (dịch vụ) bao gồm giá trị: **urn:oasis:names:tc:ebxml-msg:service**;
 - Một phần tử **Action** (hành động) bao gồm giá trị **Pong**;
- Một phần tử **MessageData** (dữ liệu thông điệp) bao gồm:
 - Một phần tử **RefToMessageId** xác định thông điệp Ping MSH.
- Một phần tử **Signature** (chữ ký) [XMLDSIG] (xem phần 4.1.1).

Một ví dụ về Pong:

...*Các tiêu đề truyền*

```

SOAPAction: "ebXML"
Content-Type: text/xml
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
    xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/

```

```

http://www.oasis-open.org/committees/ebxml-
msg/schema/envelope.xsd">
<SOAP:Header xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
header-2_0.xsd"
xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
header-2_0.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
header-2_0.xsd">
  <eb:MessageHeader eb:version="2.0" SOAP:mustUnderstand="1">
    <eb:From> <eb:PartyId>urn:duns:912345678</eb:PartyId> </eb:From>
    <eb:To> <eb:PartyId>urn:duns:123456789</eb:PartyId> </eb:To>
    <eb:CPAId>20001209-133003-28572</eb:CPAId>
    <eb:ConversationId>20010215-111213-28572</eb:ConversationId>
    <eb:Service>urn:oasis:names:tc:ebxml-msg:service</eb:Service>
    <eb:Action>Pong</eb:Action>
    <eb:MessageData>
    <eb:MessageId>20010215-111213-395884@example2.com</eb:MessageId>
    <eb:Timestamp>2001-02-15T11:12:13</eb:Timestamp>
    <eb:RefToMessageId>20010215-111212-28572@example.com</eb:RefToMessageId>
    </eb:MessageData>
  </eb:MessageHeader>
</SOAP:Header>
<SOAP:Body/>
</SOAP:Envelope>

```

CHÚ THÍCH: Ví dụ trên chỉ ra một cấu trúc MIME phi-multipart MIME structure.

8.3 Các xem xét về an ninh

Các bên nhận một thông điệp **Ping** MSH NÊN luôn phản hồi tới thông điệp đó. Tuy nhiên, có một rủi ro một vài bên có thể sử dụng thông điệp **Ping** MSH để xác định sự tồn tại của Trình quản lý dịch vụ thông điệp như một phần của một cuộc tấn công an ninh tới MSH đó. Vì vậy, Người nhận của một MSH Ping CÓ THỂ lược bỏ thông điệp nếu chúng xem xét người gửi của thông điệp nhận được là trái phép hoặc là một phần của một số cuộc tấn công.

Quá trình quyết định kết quả trong giai đoạn này là sự thực thi độc lập.

9 Môđun MessageOrder (Thứ tự thông điệp)

Môđun **MessageOrder** (thứ tự thông điệp) cho phép các thông điệp có mặt tới *To Party* theo một thứ tự cụ thể. Điều này được thực hiện thông qua việc sử dụng phần tử **MessageOrder** (thứ tự thông điệp). Việc truyền thông điệp tin cậy PHẢI được sử dụng khi phần tử **MessageOrder** (thứ tự thông điệp) có mặt.

Môđun **MessageOrder** (thứ tự thông điệp) PHẢI chỉ được sử dụng khi liên kết với ebXML, việc truyền môđun thông điệp tin cậy (phần 6) với một giản đồ Once-And-Only-Once (phần 6.6). Nếu một trình tự được gửi và một thông điệp lỗi nhận tại MSH của *To Party*, tất cả các thông điệp tiếp theo cũng lỗi khi xuất hiện trên ứng dụng *To Party* (xem thuộc tính **status** phần 7.1.1).

9.1 Phần tử MessageOrder (thứ tự thông điệp)

Phần tử **MessageOrder** (thứ tự thông điệp) là một đuôi mở rộng TỰY CHỌN cho **Header** SOAP yêu cầu việc duy trì trình tự thông điệp trong hội thoại này.

TCVN ISO/TS 15000-2 : 2007

Phần tử **MessageOrder** (thứ tự thông điệp) bao gồm:

- Một thuộc tính **id** (xem phần 2.3.7);
- Một thuộc tính **version** (phiên bản) (xem phần 2.3.8);
- Một thuộc tính **mustUnderstand** của SOAP với giá trị là "1" (xem phần 2.3.9);
- Một phần tử **SequenceNumber** (số trình tự).

Khi phần tử **MessageOrder** (thứ tự thông điệp) có mặt, **DuplicateElimination** (loại trừ sao chép) cũng PHẢI có mặt và **SyncReply** (trả lời đồng bộ) KHÔNG PHẢI có mặt.

9.1.1 Phần tử SequenceNumber (số trình tự)

Phần tử **SequenceNumber** (số trình tự) YÊU CẦU chỉ ra trình tự một **MSH nhận** PHẢI xử lý các thông điệp. **SequenceNumber**(số trình tự) là duy nhất trong **ConversationId** (id hội thoại) và MSH. MSH của From Party và MSH của To Party đặt mỗi **SequenceNumber** (số trình tự) độc lập như **MSH gửi** trong **ConversationId** (id hội thoại). Nó được đặt là 0 ở thông điệp đầu tiên từ MSH đó trong một hội thoại và sau đó tăng lên một đối với mỗi thông điệp được gửi tiếp theo.

Một MSH nhận một thông điệp cùng với phần tử **SequenceNumber** (số trình tự) KHÔNG PHẢI truyền thông điệp tới một ứng dụng đến khi tất cả các thông điệp một **SequenceNumber** (số trình tự) thấp hơn đã truyền tới ứng dụng đó.

Nếu sự thực thi đó xác định giới hạn đối với các thông điệp ngoài chuỗi được lưu trữ trong phạm vi, khi đó **MSH nhận** đó PHẢI chỉ ra một lỗi truyền tới **MSH gửi** cùng với **errorCode** (mã lỗi) đặt là **DeliveryFailure** và **severity** đặt là **Error** (xem phần 4.1.5).

Phần tử **SequenceNumber** (số trình tự) là một giá trị nguyên được gia tăng bởi **MSH gửi** (ví dụ: 0, 1, 2, 3, 4...) đối với mỗi ứng dụng thông điệp đã chuẩn bị được gửi bởi MSH đó trong phạm vi **ConversationId**. Giá trị tiếp theo của 99999999 trong sự gia tăng là "0". Giá trị của **SequenceNumber** (số trình tự) bao gồm các số ASCII trong dải 0-99999999. Trong các trường hợp sau đây, **SequenceNumber** (số trình tự) lấy giá trị "0":

1. thông điệp đầu tiên từ **MSH gửi** trong phạm vi hội thoại;
2. thông điệp đầu tiên sau việc đặt lại thông tin của **SequenceNumber** (số trình tự) bởi **MSH gửi**;
3. thông điệp đầu tiên sau khi quay vòng (giá trị tiếp theo sau 99999999).

Phần tử **SequenceNumber** (số trình tự) có một thuộc tính đơn, **status**. Thuộc tính này là một kiểu số đếm, nó PHẢI có một trong các giá trị sau đây:

- **Reset –SequenceNumber** (số trình tự) được thiết lập lại như được chỉ ra trong 1 hoặc 2 ở trên;
- **Continue –SequenceNumber** (số trình tự) tiếp tục theo trình tự (bao gồm 3 ở trên).

Khi **SequenceNumber** (số trình tự) được đặt là "0" do 1 hoặc 2 ở trên, *MSH gửi* PHẢI đặt thuộc tính **status** của thông điệp là **Reset**. Trong tất cả các trường hợp khác, gồm 3 ở trên, thuộc tính **status** PHẢI được đặt là **Continue**. Giá trị mặc định của thuộc tính **status** là **Continue**.

Một *MSH gửi* PHẢI đợi trước khi việc thiết lập lại **SequenceNumber**(số trình tự) của một hội thoại cho tới khi nó có xác nhận được của tất cả các thông điệp được gửi trước đó đối với hội thoại đó. Chỉ khi tất cả các thông điệp đã gửi được giải thích, thì có thể *MSH gửi* đặt lại **SequenceNumber** (số trình tự).

9.1.2 Ví dụ về MessageOrder (Thứ tự thông điệp)

Một ví dụ về của phần tử MessageOrder cho trước dưới đây:

```
<eb:MessageOrder eb:version="2.0" SOAP:mustUnderstand="1">
  <eb:SequenceNumber>00000010</eb:SequenceNumber>
</eb:MessageOrder>
```

9.2 Sự tương tác của phần tử MessageOrder (Thứ tự thông điệp)

Đối với tiêu chuẩn này, phần tử **MessageOrder** (thứ tự thông điệp) KHÔNG PHẢI có mặt cùng với phần tử **SyncReply**. Nếu hai phần tử này cùng nhận được trong một thông điệp, thì *MSH nhận* NÊN báo cáo một lỗi (xem phần 4.1.5) với **errorCode** (mã lỗi) đặt là **Inconsistent** và **severity** đặt là **Error**.

10 Môđun Multi-Hop (đa bước truyền)

Multi-hop (đa bước truyền) là quá trình truyền thông điệp thông qua một hoặc nhiều nút hoặc MSH trung gian. Một nút trung gian là nút hoặc MSH nào đó mà ở đó thông điệp nhận được, nhưng không PHẢI là *MSH nhận* hoặc *gửi*. Nút này được gọi là nút trung gian.

Các nút trung gian có thể sử dụng cho mục đích lưu trữ và chuyển tiếp hoặc liên quan trong một số hoạt động xử lý như dịch vụ tem thời gian của bên thứ ba được chứng thực. Đối với mục đích của tiêu chuẩn này, các nút trung gian chỉ được xem như các thực thể lưu trữ và chuyển tiếp. Các nút trung gian CÓ THỂ liên quan đến việc gỡ bỏ và bổ sung các phần tử đuôi mở rộng của SOAP hoặc các môđun được nhằm tới nút **Next** của SOAP hoặc **NextMSH**. Các quy tắc SOAP quy định gỡ bỏ bất kỳ phần tử hoặc môđun nhằm vào nút **Next** của SOAP. Nếu phần tử hoặc môđun đó cần thiết để tiếp tục xuất hiện trên thông điệp SOAP được đến nút **Next** của SOAP hoặc **NextMSH** trong tiêu chuẩn này, nó PHẢI được áp dụng lại. Việc xoá bỏ và bổ sung các phần tử hoặc môđun đưa ra các khó khăn tiềm tàng cho các thông điệp ebXML được ký. Bất kỳ nút hoặc MSH trung gian KHÔNG PHẢI thay đổi, định dạng hoặc thực hiện một cách sửa đổi nào trên bất kỳ phần tử nào không nhằm tới nút trung gian đó. Bất kỳ thay đổi nào như vậy có thể làm mất tính hiệu lực của chữ ký đó.

10.1 Việc truyền thông điệp tin cậy Multi-hop (đa bước truyền)

Việc truyền thông điệp tin cậy Multi-hop (hop-to-hop) được thực hiện thông qua việc sử dụng phần tử **AckRequested** (yêu cầu báo nhận) (phần 6.3.1) và một *thông điệp báo nhận* bao gồm Một phần tử **Acknowledgment** (phần 6.3.1.4) mỗi phần tử cùng với **actor** SOAP của **MSHNext** (phần 2.3.10) giữa *MSH gửi* và *MSH nhận*.

Điều này CÓ THỂ được sử dụng trong các trường hợp multi-hop lưu trữ và chuyển tiếp.

Việc sử dụng việc loại trừ bản sao giống hệt không được yêu cầu đối với các nút trung gian. Do việc loại bỏ bản sao giống hệt bởi MSH trung gian có thể giao tiếp với việc truyền thông điệp tin cậy của các Retry End-to-End, MSH trung gian đó PHẢI biết nó là một nút trung gian và KHÔNG PHẢI thực hiện các nhiệm vụ loại trừ bản sao giống hệt.

Tại thời điểm này, các giá trị của **Retry** và **RetryInterval** giữa các MSH trung gian vẫn thi hành cụ thể. Việc truyền thông điệp tin cậy, xem phần 6.4.

10.1.1 Ví dụ minh họa về AckRequested (yêu cầu báo nhận)

Một ví dụ về phần tử **AckRequested** (yêu cầu báo nhận) nhằm tới **NextMSH** cho trước:

```
<eb:AckRequested SOAP:mustUnderstand="1" eb:version="2.0" eb:signed="false"
  SOAP:actor="urn:oasis:names:tc:ebxml-msg:actor:nextMSH"/>
```

Trong ví dụ trước, nút MSH ebXML tiếp theo yêu cầu một *thông điệp báo nhận* (xem phần 2.3.10) trong thông điệp. Phần tử **Acknowledgment** (báo nhận) được tạo PHẢI nhằm tới nút MSH ebXML tiếp theo dọc theo đường dẫn thông điệp ngược (*MSH gửi*) có sử dụng **actor** SOAP với giá trị là **NextMSH** (phần 2.3.10).

Bất kỳ nút trung gian nào nhận một **AckRequested** (yêu cầu báo nhận) cùng với actor SOAP của NextMSH PHẢI gỡ bỏ phần tử **AckRequested** (yêu cầu báo nhận) trước khi chuyển tiếp đến MSHNext. Bất kỳ nút trung gian nào CÓ THỂ chèn Một phần tử **AckRequested** (yêu cầu báo nhận) đơn vào **Header** SOAP cùng với actor SOAP của NextMSH. KHÔNG PHẢI có hai phần tử **AckRequested** (yêu cầu báo nhận) nhằm tới MSHNext.

Khi phần tử **SyncReply** có mặt, Một phần tử **AckRequested** (yêu cầu báo nhận) cùng với actor SOAP của NextMSH KHÔNG PHẢI có mặt. Nếu phần tử **SyncReply** không có mặt, nút trung gian đó CÓ THỂ trả lại *Thông điệp báo nhận* của nút trung gian đồng thời với giao thức truyền đồng bộ. Nếu hai phần tử này được nhận trong cùng thông điệp, MSH *nhận* NÊN báo cáo một lỗi (xem phần 4.1.5) là **errorCode** (mã lỗi) đặt là **Inconsistent** và **severity** đặt là **Error**.

10.1.2 Ví dụ về Acknowledgment (Báo nhận)

Một ví dụ về phần tử **Acknowledgment** (báo nhận) nhằm tới **NextMSH** cho trước dưới đây:

```
<eb:Acknowledgment SOAP:mustUnderstand="1" eb:version="2.0"
  SOAP:actor="urn:oasis:names:tc:ebxml-msg:actor:nextMSH">
  <eb:Timestamp>2001-03-09T12:22:30</eb:Timestamp>
  <eb:RefToMessageId>323210:e52151ec74:-7ffc@xtacy</eb:RefToMessageId>
  <eb:From> <eb:PartyId>uri:www.example.com</eb:PartyId> </eb:From>
</eb:Acknowledgment>
```

10.1.3 Báo nhận Multi-Hop

CÓ THỂ có hai phần tử **AckRequested** (yêu cầu báo nhận) trên cùng một thông điệp. Một **Acknowledgment** PHẢI được gửi cho mỗi **AckRequested** có sử dụng một thuộc tính **actor** của

SOAP đồng dạng như phần tử **AckRequested** (yêu cầu báo nhận). Nếu MSH *nhận* là MSH của *To Party*, thì xem phần 6.5.2. Nếu MSH *nhận* là MSH của *To Party* và có Một phần tử **AckRequested** (yêu cầu báo nhận) nhằm tới MSHNext (MSH của *To Party* đóng cả hai vai trò), thì thực hiện cả hai thủ tục (phần này và phần 6.5.2) để tạo các *Thông điệp báo nhận*. Điều này CÓ THỂ yêu cầu gửi hai phần tử **Acknowledgment**, có thể trên cùng thông điệp, Một phần tử nhằm tới MSHNext và Một phần tử nhằm tới MSH của *To Party*.

CÓ THỂ có nhiều Phần tử **Acknowledgements**, trên cùng thông điệp hoặc trên các thông điệp khác nhau, việc trả lại từ MSHNext hoặc từ MSH của *To Party*. Một MSH hỗ trợ Multi-hop PHẢI phân biệt trên cơ sở **actor**, mà **Acknowledgment** đang gửi trả lại và hoạt động phù hợp với điều đã được nhắc đến.

Nếu đây là một *thông điệp báo nhận* như xác định trong phần 8 thì:

1. tìm kiếm thông điệp trong *bộ lưu trữ lâu dài* cùng với một **MessageId** the same as giá trị của **RefToMessageId** on the received Thông điệp;
2. nếu một thông điệp được tìm thấy trong *bộ lưu trữ lâu dài* thì đánh dấu thông điệp vẫn tồn tại khi được truyền.

Nếu Một phần tử **AckRequested** (yêu cầu báo nhận) có mặt (không PHẢI là một *thông điệp báo nhận*) thì tạo ra một *thông điệp báo nhận* phản hồi (điều này có thể như là một phần của một thông điệp khác).

MSH *nhận* KHÔNG PHẢI gửi một *thông điệp báo nhận* đến khi thông điệp được tiếp tục tồn tại hoặc được gửi tới MSHNext.

10.1.4 Ký hiệu báo nhận Multi-Hop (đa bước truyền)

Khi một *thông điệp báo nhận* trung gian đã ký được yêu cầu (như là một *thông điệp báo nhận* đã ký cùng với một **actor** SOAP của NextMSH), nó Phải được tự gửi và không được gói cùng với bất kỳ thông điệp khác. Phần tử **Signature** (chữ ký) [XMLDSIG] của chữ ký XML cùng với **Transforms**, như được mô tả trong phần 4.1.3, PHẢI loại trừ phần tử **Acknowledgment** này. Để gửi một *thông điệp báo nhận* đã ký cùng với **actor** SOAP của NextMSH, tạo một thông điệp không có vùng mang thông tin, bao gồm Một phần tử **Acknowledgment** đơn (xem phần 6.3.2.6) và Một phần tử **Signature** (chữ ký) [XMLDSIG] cùng với **Transforms** sau đây:

```
<Transforms>
  <Transform Algorithm="http://www.w3.org/2000/09/xmlsig#enveloped-
signature"/>
  <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
</Transforms>
```

10.1.5 Xem xét an ninh của Multi-Hop (đa bước truyền)

Việc truyền *thông điệp* SOAP cho phép các nút trung gian bổ sung hoặc gỡ bỏ các phần tử được nhằm tới nút trung gian đó.

TCVN ISO/TS 15000-2 : 2007

Điều này bao gồm các mâu thuẫn tiềm ẩn với các chữ ký end-to-end do thay đổi yếu nhất trong bất kỳ ký tự của **Envelope** SOAP (đường bao SOAP) hoặc tới một vùng mang thông tin PHẢI hết hiệu lực **ds:Signature** bởi việc thay đổi bản tóm lược được tính toán.

Các nút trung gian KHÔNG PHẢI bổ sung hoặc gỡ bỏ các phần tử trừ khi chúng bao gồm **actor** SOAP của **next** hoặc **nextMSH**. Các nút trung gian KHÔNG PHẢI xáo trộn khoảng trống trắng – Các ký tự kết thúc dòng (CR/LF), tabs, các khoảng trống, v..v. – Bên ngoài các phần tử được bổ sung và gỡ bỏ đó.

10.2 Sắp xếp thứ tự thông điệp và Multi-Hop (đa bước truyền)

Các nút MSH trung gian KHÔNG PHẢI tham gia trong quá trình xử lý thông điệp Order như được quy định trong phần 7.

Chương III: Các phụ lục chuẩn

Phụ lục A

Giản đồ các phần tử đuôi mở rộng của SOAP của ebXML

Ban kỹ thuật về truyền thông điệp ebXML của OASIS đã cung cấp một phiên bản giản đồ đường bao SOAP 1.1 quy định việc sử dụng từ vựng giản đồ phù hợp với quy định KHUYẾN CÁO giản đồ XML của W3C [XMLSchema].

SOAP1.1- <http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd>

Cần thiết tạo một giản đồ từ vựng thuộc tính XLINK [XLINK] phù hợp với KHUYẾN CÁO giản đồ XML của W3C [XMLSchema]. Giản đồ này được tham chiếu từ giản đồ các phần tử đuôi mở rộng SOAP của ebXML và sẵn có trên URL:

Xlink - <http://www.oasis-open.org/committees/ebxml-msg/schema/xlink.xsd>

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Some parsers may require explicit declaration of
      xmlns:xml="http://www.w3.org/XML/1998/namespace".
      In that case, a copy of this schema augmented with the above declaration
      should be cached and used
      for the purpose of schema validation on ebXML messages. -->
<schema targetNamespace="http://www.oasis-open.org/committees/ebxml-
      msg/schema/msg-header-2_0.xsd"
      xmlns:tns="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-
      2_0.xsd"
      xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
      xmlns:xlink="http://www.w3.org/1999/xlink"
      xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns="http://www.w3.org/2001/XMLSchema"
      elementFormDefault="qualified"
      attributeFormDefault="qualified"
      version="2.0">
<import namespace="http://www.w3.org/2000/09/xmldsig#"
      schemaLocation="http://www.w3.org/TR/xmldsig-core/xmldsig-core-schema.xsd"/>
<import namespace="http://www.w3.org/1999/xlink"
      schemaLocation="http://www.oasis-open.org/committees/ebxml-
      msg/schema/xlink.xsd"/>
<import namespace="http://schemas.xmlsoap.org/soap/envelope/"
      schemaLocation="http://www.oasis-open.org/committees/ebxml-
      msg/schema/envelope.xsd"/>
<import namespace="http://www.w3.org/XML/1998/namespace"
      schemaLocation="http://www.w3.org/2001/03/xml.xsd"/>
<!-- MANIFEST, for use in soap:Body element -->
<element name="Manifest">
  <complexType>
    <sequence>
      <element ref="tns:Reference" maxOccurs="unbounded"/>
      <any namespace="##other" processContents="lax" minOccurs="0"
          maxOccurs="unbounded"/>
    </sequence>
    <attributeGroup ref="tns:bodyExtension.grp"/>
  </complexType>
</element>
<element name="Reference">
  <complexType>
    <sequence>
      <element ref="tns:Schema" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
```

```

    <element ref="tns:Description" minOccurs="0" maxOccurs="unbounded"/>
    <any namespace="##other" processContents="lax" minOccurs="0"
        maxOccurs="unbounded"/>
</sequence>
<attribute ref="tns:id"/>
<attribute ref="xlink:type" fixed="simple"/>
<attribute ref="xlink:href" use="required"/>
<attribute ref="xlink:role"/>
<anyAttribute namespace="##other" processContents="lax"/>
</complexType>
</element>
<element name="Schema">
    <complexType>
        <attribute name="location" type="anyURI" use="required"/>
        <attribute name="version" type="tns:non-empty-string"/>
    </complexType>
</element>
<!-- MESSAGEHEADER, for use in soap:Header element -->
<element name="MessageHeader">
    <complexType>
        <sequence>
            <element ref="tns:From"/>
            <element ref="tns:To"/>
            <element ref="tns:CPAId"/>
            <element ref="tns:ConversationId"/>
            <element ref="tns:Service"/>
            <element ref="tns:Action"/>
            <element ref="tns:MessageData"/>
            <element ref="tns:DuplicateElimination" minOccurs="0"/>
            <element ref="tns:Description" minOccurs="0" maxOccurs="unbounded"/>
            <any namespace="##other" processContents="lax" minOccurs="0"
                maxOccurs="unbounded"/>
        </sequence>
        <attributeGroup ref="tns:headerExtension.grp"/>
    </complexType>
</element>
<element name="CPAId" type="tns:non-empty-string"/>
<element name="ConversationId" type="tns:non-empty-string"/>
<element name="Service">
    <complexType>
        <simpleContent>
            <extension base="tns:non-empty-string">
                <attribute name="type" type="tns:non-empty-string"/>
            </extension>
        </simpleContent>
    </complexType>
</element>
<element name="Action" type="tns:non-empty-string"/>
<element name="MessageData">
    <complexType>
        <sequence>
            <element ref="tns:MessageId"/>
            <element ref="tns:Timestamp"/>
            <element ref="tns:RefToMessageId" minOccurs="0"/>
            <element ref="tns:TimeToLive" minOccurs="0"/>
        </sequence>
    </complexType>
</element>
<element name="MessageId" type="tns:non-empty-string"/>
<element name="TimeToLive" type="dateTime"/>
<element name="DuplicateElimination">
</element>
<!-- SYNC REPLY, for use in soap:Header element -->
<element name="SyncReply">

```

```

<complexType>
  <sequence>
    <any namespace="##other" processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </sequence>
  <attributeGroup ref="tns:headerExtension.grp"/>
  <attribute ref="soap:actor" use="required"/>
</complexType>
</element>
<!-- MESSAGE ORDER, for use in soap:Header element -->
<element name="MessageOrder">
  <complexType>
    <sequence>
      <element ref="tns:SequenceNumber"/>
      <any namespace="##other" processContents="lax" minOccurs="0"
        maxOccurs="unbounded"/>
    </sequence>
    <attributeGroup ref="tns:headerExtension.grp"/>
  </complexType>
</element>
<element name="SequenceNumber" type="tns:sequenceNumber.type"/>
<!-- ACK REQUESTED, for use in soap:Header element -->
<element name="AckRequested">
  <complexType>
    <sequence>
      <any namespace="##other" processContents="lax" minOccurs="0"
        maxOccurs="unbounded"/>
    </sequence>
    <attributeGroup ref="tns:headerExtension.grp"/>
    <attribute ref="soap:actor"/> 2273
    <attribute name="signed" type="boolean" use="required"/>
  </complexType>
</element>
<!-- ACKNOWLEDGMENT, for use in soap:Header element -->
<element name="Acknowledgment">
  <complexType>
    <sequence>
      <element ref="tns:Timestamp"/>
      <element ref="tns:RefToMessageId"/>
      <element ref="tns:From" minOccurs="0"/>
      <element ref="ds:Reference" minOccurs="0" maxOccurs="unbounded"/>
      <any namespace="##other" processContents="lax" minOccurs="0"
        maxOccurs="unbounded"/>
    </sequence>
    <attributeGroup ref="tns:headerExtension.grp"/>
    <attribute ref="soap:actor"/>
  </complexType>
</element>
<!-- ERROR LIST, for use in soap:Header element -->
<element name="ErrorList">
  <complexType>
    <sequence>
      <element ref="tns:Error" maxOccurs="unbounded"/>
      <any namespace="##other" processContents="lax" minOccurs="0"
        maxOccurs="unbounded"/>
    </sequence>
    <attributeGroup ref="tns:headerExtension.grp"/>
    <attribute name="highestSeverity" type="tns:severity.type" use="required"/>
  </complexType>
</element>
<element name="Error">
  <complexType>
    <sequence>
      <element ref="tns:Description" minOccurs="0"/>

```

```

    <any namespace="##other" processContents="lax" minOccurs="0"
        maxOccurs="unbounded"/>
</sequence>
<attribute ref="tns:id"/>
<attribute name="codeContext" type="anyURI"
    default="urn:oasis:names:tc:ebxml-msg:service:errors"/>
<attribute name="errorCode" type="tns:non-empty-string" use="required"/>
<attribute name="severity" type="tns:severity.type" use="required"/>
<attribute name="location" type="tns:non-empty-string"/>
<anyAttribute namespace="##other" processContents="lax"/>
</complexType>
</element>
<!-- STATUS RESPONSE, for use in soap:Body element -->
<element name="StatusResponse">
    <complexType>
        <sequence>
            <element ref="tns:RefToMessageId"/>
            <element ref="tns:Timestamp" minOccurs="0"/>
            <any namespace="##other" processContents="lax" minOccurs="0"
                maxOccurs="unbounded"/>
        </sequence>
        <attributeGroup ref="tns:bodyExtension.grp"/>
        <attribute name="messageStatus" type="tns:messageStatus.type"
            use="required"/>
    </complexType>
</element>
<!-- STATUS REQUEST, for use in soap:Body element -->
<element name="StatusRequest">
    <complexType>
        <sequence>
            <element ref="tns:RefToMessageId"/>
            <any namespace="##other" processContents="lax" minOccurs="0"
                maxOccurs="unbounded"/>
        </sequence>
        <attributeGroup ref="tns:bodyExtension.grp"/>
    </complexType>
</element>
<!-- COMMON TYPES -->
<complexType name="sequenceNumber.type">
    <simpleContent>
        <extension base="nonNegativeInteger">
            <attribute name="status" type="tns:status.type" default="Continue"/>
        </extension>
    </simpleContent>
</complexType>
<simpleType name="status.type">
    <restriction base="NMTOKEN">
        <enumeration value="Reset"/>
        <enumeration value="Continue"/>
    </restriction>
</simpleType>
<simpleType name="messageStatus.type">
    <restriction base="NMTOKEN">
        <enumeration value="Unauthorized"/>
        <enumeration value="NotRecognized"/>
        <enumeration value="Received"/>
        <enumeration value="Processed"/>
        <enumeration value="Forwarded"/>
    </restriction>
</simpleType>
<simpleType name="non-empty-string">
    <restriction base="string">
        <minLength value="1"/>
    </restriction>

```



```

</simpleType>
<simpleType name="severity.type">
  <restriction base="NMTOKEN">
    <enumeration value="Warning"/>
    <enumeration value="Error"/>
  </restriction>
</simpleType>
<!-- COMMON ATTRIBUTES and ATTRIBUTE GROUPS -->
<attribute name="id" type="ID"/>
<attribute name="version" type="tns:non-empty-string"/>
<attributeGroup name="headerExtension.grp">
  <attribute ref="tns:id"/>
  <attribute ref="tns:version" use="required"/>
  <attribute ref="soap:mustUnderstand" use="required"/>
  <anyAttribute namespace="##other" processContents="lax"/>
</attributeGroup>
<attributeGroup name="bodyExtension.grp">
  <attribute ref="tns:id"/>
  <attribute ref="tns:version" use="required"/>
  <anyAttribute namespace="##other" processContents="lax"/>
</attributeGroup>
<!-- COMMON ELEMENTS -->
<element name="PartyId">
  <complexType>
    <simpleContent>
      <extension base="tns:non-empty-string">
        <attribute name="type" type="tns:non-empty-string"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="To">
  <complexType>
    <sequence>
      <element ref="tns:PartyId" maxOccurs="unbounded"/>
      <element name="Role" type="tns:non-empty-string" minOccurs="0"/>
    </sequence>
  </complexType>
</element>
<element name="From">
  <complexType>
    <sequence>
      <element ref="tns:PartyId" maxOccurs="unbounded"/>
      <element name="Role" type="tns:non-empty-string" minOccurs="0"/>
    </sequence>
  </complexType>
</element>
<element name="Description">
  <complexType>
    <simpleContent>
      <extension base="tns:non-empty-string">
        <attribute ref="xml:lang" use="required"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="RefToMessageId" type="tns:non-empty-string"/>
<element name="Timestamp" type="dateTime"/>
</schema>

```

Phụ lục B

Quy định giao thức truyền thông

B.1 Giới thiệu

Một trong các mục tiêu của quy chuẩn này là thiết kế một thông điệp xử lý dịch vụ có thể sử dụng các giao thức mạng và truyền tải ứng dụng. Các giao thức này được xem như là sóng mang các thông điệp ebXML và cung cấp các dịch vụ cần thiết để thực hiện một trao đổi thông điệp ebXML hoàn thành giữa 2 bên. HTTP, FTP, dịch vụ thông điệp Java (JMS) và SMTP là các ví dụ về giao thức truyền tải ứng dụng. TCP, FTP/LU6.2 là các ví dụ về giao thức truyền mạng. Các giao thức truyền tải thực hiện hỗ trợ thông tin dữ liệu, cách xử lý và quá trình xử lý và báo lỗi dưới nhiều hình thức. Chẳng hạn, nó thường gửi các dữ liệu nhị phân qua giao thức HTTP. Tuy nhiên, trong trường hợp của SMTP, nó lại thường “mã hoá” các dữ liệu nhị phân qua giao thức trình bày 7-bit. HTTP cũng có thể thực hiện các trao đổi thông điệp *đồng bộ* hoặc *không đồng bộ*, trong khi đó có khả năng các trao đổi thông điệp xảy ra qua SMTP PHẢI không đồng bộ với nhau. Phần này mô tả các chi tiết kỹ thuật cần thiết để thực hiện Dịch vụ Xử lý thông điệp ebXML lý thuyết này qua các giao thức truyền tải cụ thể.

Phần này ghi rõ các đóng gói giao thức truyền thông và chi tiết kỹ thuật để thực hiện các thông điệp Dịch vụ *thông điệp ebXML* dành cho các giao thức truyền thông sau:

- giao thức Truyền tải Siêu văn bản [RFC2616], dưới hình thức truyền tải không đồng bộ và đồng bộ;
- giao thức Truyền tải Thư Đơn giản [RFC2821], chỉ dưới hình thức truyền tải không đồng bộ.

B.2 HTTP

B.2.1 Mức giao thức HTTP nhỏ nhất

Bản Giao thức Truyền tải Siêu văn bản 1.1 [RFC2616] là mức giao thức nhỏ nhất PHẢI được sử dụng.

B.2.2 Gửi các thông điệp dịch vụ ebXML qua giao thức HTTP

Cho dù có tồn tại một số phương pháp yêu cầu HTTP, quy chuẩn này chỉ định nghĩa cách sử dụng các yêu cầu HTTP POST để gửi các thông điệp Dịch vụ Thông điệp ebXML qua HTTP. Tính đồng nhất của MSH ebXML (ví dụ, ebxmlhandler) có thể nằm trong yêu cầu của HTTP POST:

Trước khi gửi qua HTTP, một thông điệp ebXML PHẢI được định dạng theo Quy chuẩn Dịch vụ Thông điệp ebXML. Hơn nữa, các thông điệp PHẢI tuân theo các giới hạn mẫu quy tắc MIME cụ thể HTTP được ghi rõ trong phần 19.4 của quy chuẩn RFC 2616 [RFC2616].

Về bản chất, giao thức HTTP hỗ trợ các dữ liệu 8-bit và nhị phân. Vì vậy, mã hoá truyền tải cho các phần này trong Thông điệp Dịch vụ ebXML trước khi gửi qua HTTP là OPTIONAL.

Tuy nhiên, mã hoá truyền tải nội dung của các phần này (ví dụ, sử dụng hệ thống mã hoá base 64) có thể xảy ra theo quy chuẩn này.

Dưới đây là các nguyên tắc lập một thông điệp HTTP có bao gồm Thông điệp Dịch vụ ebXML:

- Content-type: Tiêu đề MIME đa phần/ liên quan với các tham số kết hợp từ Phong bì Thông điệp Dịch vụ ebXML PHẢI xuất hiện dưới dạng tiêu đề http;
- tất cả các tiêu đề MIME khác cấu thành Phong bì *thông điệp ebXML* cũng PHẢI trở thành tiêu đề của http;
- trường tiêu đề SOAPAction HTTP bắt buộc cũng PHẢI nằm trong tiêu đề HTTP và có thể có giá trị "ebXML" SOAPAction: "ebXML";
- các tiêu đề khác có ngữ nghĩa học được xác định theo các quy chuẩn MIME như Mã hóa Truyền tải Nội dung không được xuất hiện dưới dạng các tiêu đề HTTP. Đặc biệt, tiêu đề "Bản MIME: 1.0" không được xuất hiện dưới dạng một tiêu đề HTTP. Tuy nhiên, các tiêu đề như MIME theo quy chuẩn HTTP được xác định theo HTTP 1.1 có thể được sử dụng theo ngữ nghĩa học được xác định trong quy chuẩn HTTP.

Toàn bộ phần Thông điệp Dịch vụ ebXML theo Phong bì Thông điệp ebXML, bao gồm chuỗi đường biên MIME PHẢI cấu thành HTTP (HTTP entity body). Phần này bao gồm Phong bì SOAP và các phần ebXML hợp thành và các phần đính kèm trong đó có các chuỗi đường biên MIME kéo dài (trailing MIME boundary strings).

Ví dụ dưới đây PHẢI cho thấy trường hợp của một thông điệp Dịch vụ HTTP POST ebXML:

```
POST /servlet/ebXMLhandler HTTP/1.1
Host: www.example2.com
SOAPAction: "ebXML"
Content-type: multipart/related; boundary="Boundary"; type="text/xml";
            start="<ebxhmheader111@example.com>"

--Boundary
Content-ID: <ebxhmheader111@example.com>
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
    header-2_0.xsd"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
    http://www.oasis-open.org/committees/ebxml-
    msg/schema/envelope.xsd
    http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
    header-2_0.xsd
    http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
    header-2_0.xsd">
<SOAP:Header>
  <eb:MessageHeader SOAP:mustUnderstand="1" eb:version="2.0">
    <eb:From>
      <eb:PartyId>urn:duns:123456789</eb:PartyId>
    </eb:From>
    <eb:To>
      <eb:PartyId>urn:duns:912345678</eb:PartyId>
    </eb:To>
    <eb:CPAId>20001209-133003-28572</eb:CPAId>
    <eb:ConversationId>20001209-133003-28572</eb:ConversationId>
```

```
<eb:Service>urn:services:SupplierOrderProcessing</eb:Service>
<eb:Action>NewOrder</eb:Action>
<eb:MessageData>
  <eb:MessageId>20001209-133003-28572@example.com</eb:MessageId>
  <eb:Timestamp>2001-02-15T11:12:12</eb:Timestamp>
</eb:MessageData>
</eb:MessageHeader>
</SOAP:Header>
<SOAP:Body>
  <eb:Manifest eb:version="2.0">
    <eb:Reference xlink:href="cid:ebxmlpayload111@example.com"
      xlink:role="XLinkRole"  xlink:type="simple">
      <eb:Description xml:lang="en-US">Purchase Order 1</eb:Description>
    </eb:Reference>
  </eb:Manifest>
</SOAP:Body>
</SOAP:Envelope>

--Boundary
Content-ID: <ebxmlpayload111@example.com>
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>
<purchase_order>
  <po_number>1</po_number>
  <part_number>123</part_number>
  <price currency="USD">500.00</price>
</purchase_order>

--Boundary--
```

B.2.3 Mã trả lời HTTP

Nhìn chung, các ngữ nghĩa học của kết nối qua HTTP được ghi trong [RFC2616] PHẢI như sau để quay trở lại các mã trả lời mức HTTP. Một mã 2xx PHẢI được quay trở lại khi thông điệp Đã gửi HTTP được nhận thành công qua chủ thể HTTP đang nhận. Tuy nhiên, dưới đây chúng ta có thể thấy trường hợp ngoại lệ của các điều kiện lỗi SOAP. Tương tự, các mã HTTP khác có dải 3xx, 4xx, 5xx cũng có thể quay trở lại các điều kiện tương đương với chúng. Tuy nhiên, các điều kiện lỗi xảy ra trong quá trình xử lý một thông điệp Dịch vụ ebXML PHẢI được báo cáo sử dụng cơ chế lỗi theo Quy chuẩn Dịch vụ *thông điệp ebXML* (xem phần 4.1.5).

B.2.4 Các điều kiện lỗi SOAP và trao đổi đồng bộ

Quy chuẩn SOAP 1.1 nêu như sau:

“Trường hợp một lỗi SOAP xảy ra trong quá trình xử lý yêu cầu thì máy chủ SOAP HTTP PHẢI đưa ra một trả lời “Lỗi Máy chủ Nội bộ” HTTP 500 và trong đó chứa một thông điệp SOAP có Một phần tử SOAP Fault chỉ ra lỗi xử lý SOAP.”

Tuy nhiên, quy chuẩn SOAP 1.1 chỉ áp dụng cho mã trao đổi thông điệp đồng bộ qua HTTP, trong khi đó Quy chuẩn Dịch vụ *thông điệp ebXML* dành cho cả mã trao đổi thông tin đồng bộ và không đồng bộ qua HTTP.

Vì vậy, mã trao đổi thông điệp đồng bộ PHẢI theo quy chuẩn SOAP 1.1, trong khi đó Thông điệp SOAP có bao gồm Một phần tử SOAP fault chỉ ra lỗi xử lý SOAP PHẢI được quay trở lại trong trả lời HTTP bằng mã trả lời “Lỗi Máy chủ Nội bộ HTTP 500”. Khi đang sử dụng mã trao đổi thông tin không đồng bộ thì mã trả lời HTTP trong dải 2xx PHẢI được quay trở lại khi thông điệp được nhận thành công và bất kỳ một điều kiện lỗi nào (bao gồm các lỗi SOAP) cũng PHẢI được quay trở lại thông qua HTTP Post riêng.

B.2.5 Đồng bộ và không đồng bộ

Khi sử dụng một truyền tải đồng bộ, (các) thông điệp trả lời MSH PHẢI được quay trở lại trong kết nối HTTP tương tự như yêu cầu phản hồi (inbound request) bằng một mã trả lời HTTP phù hợp theo như đề cập ở trên. Khi các tham số **syncReplyMode** được thiết lập các giá trị khác **none** thì các thông điệp trả lời ứng dụng, nếu có, cũng PHẢI quay trở lại trong kết nối HTTP tương tự như yêu cầu phản hồi (inbound request) chứ không sử dụng yêu cầu HTTP Post độc lập. Nếu **syncReplyMode** có giá trị **none** thì một trả lời HTTP với mã trả lời được xác định trong phần B.2.3 nói trên và với một phần HTTP trống PHẢI được quay trở lại tương ứng với HTTP Post.

B.2.6 Điều khiển truy cập

Các công cụ (implementer) có thể bảo vệ Các trình quản lý dịch vụ thông điệp *ebXML* khỏi các truy cập trái phép bằng cách áp dụng cơ chế điều khiển truy cập. Quá trình thẩm định quyền truy cập HTTP được mô tả trong “Thẩm định quyền HTTP: Thẩm định quyền truy cập Cơ bản và Chi tiết” [RFC2617] xác định các cơ chế điều khiển truy cập được phép nhằm bảo vệ Một trình quản lý dịch vụ thông điệp *ebXML* khỏi truy cập trái phép.

Khi sử dụng Điều khiển Truy cập, các công cụ có thể hỗ trợ toàn bộ các gián đồ điều khiển truy cập được ghi rõ trong [RFC2617] trong đó bao gồm cả cơ chế Thẩm định quyền Cơ bản như được mô tả trong phần 2 của [RFC2617].

Các công cụ có sử dụng thẩm định quyền cơ bản cho điều khiển truy cập cũng PHẢI áp dụng sự an toàn giao thức truyền thông như đã được ghi rõ trong phần có tiêu đề “Tính an ninh và Sự an toàn Giao thức truyền thông” của tài liệu này.

B.2.7 An ninh và giao thức truyền thông an toàn

Một trình quản lý dịch vụ thông điệp *ebXML* có thể sử dụng mã hóa lớp truyền tải để bảo đảm tính an ninh của các thông điệp *ebXML* và các tiêu đề truyền tải HTTP. TLS quy chuẩn An toàn Lớp Truyền tải IETF [RFC2246] đưa ra các phần chi tiết kỹ thuật cụ thể và danh mục các lựa chọn cho phép thể được sử dụng bằng Các trình quản lý dịch vụ thông điệp *ebXML*.

Các trình quản lý dịch vụ thông điệp *ebXML* PHẢI có khả năng hoạt động dưới mã tương thích ngược với SSL [SSL3] như được ghi rõ trong Phụ lục E của TLS [RFC2246].

TCVN ISO/TS 15000-2 : 2007

Các trình quản lý dịch vụ thông điệp *ebXML* có thể sử dụng bất kỳ một thuật toán mã hóa và cỡ phím (key size) cho phép nào như được ghi rõ trong TLS [RFC2246]. Các trình quản lý dịch vụ thông điệp *ebXML* PHẢI ít nhất hỗ trợ được các cỡ phím và thuật toán cần thiết cho tương thích ngược với [SSL3].

Được phép sử dụng các phím/ thuật toán mã hóa 40-bit, tuy nhiên nên sử dụng các phím/ thuật toán mã hóa mạnh hơn.

Cả TLS [RFC2246] và SSL [SSL3] đều yêu cầu sử dụng các chứng nhận số máy chủ. Nó cũng đòi hỏi cả chứng nhận khách hàng dựa trên quyền thẩm định. Tất cả Các trình quản lý dịch vụ thông điệp *ebXML* đều PHẢI hỗ trợ các mô hình phân cấp và đồng đẳng (peer-to-peer) hoặc ủy thác trực tiếp.

Cả hai TLS [RFC2246] và SSL [SSL3] yêu cầu việc sử dụng các chứng chỉ số bên máy chủ. Cũng cho phép xác thực chứng chỉ bên khách. Toàn bộ Trình quản lý dịch vụ thông điệp *ebXML* PHẢI hỗ trợ các phương thức chứng thực phân tầng và peer-to-peer hoặc chứng thực trực tiếp.

B.3 SMTP

Đặc tả Giao thức truyền thư điện tử đơn giản (SMTP) [RFC 2821] thường được nhắc đến giống như thư điện tử trên internet. Đặc tả kỹ thuật này đã từng được tăng lên trong nhiều năm bởi các quy định kỹ thuật khác, đặc điểm mà xác nhận chức năng bổ xung trên và của quy định đường cơ sở này. Chúng bao gồm:

Multipurpose Internet Mail Extensions (đuôi mở rộng thư điện tử vạn năng) (MIME) [RFC 2046], [RFC 2387]

SMTP Service Extension for Authentication (đuôi mở rộng dịch vụ cho sự thẩm định quyền) [RFC 2554]

SMTP Service Extension for Secure SMTP over TLS [RFC2487] (đuôi mở rộng dịch vụ cho việc bảo vệ SMTP trên toàn bộ TLS [RFC2487])

Điển hình, sự thực hiện 2 dạng Internet Electronic Mail (thư điện tử) bao gồm 2 dạng “tác nhân” sau:

Đại diện bên truyền thông điệp (MTA): Các chương trình gửi và nhận thông điệp với MTA khác đại diện của MUA's. Máy chủ trao đổi của Microsoft là một ví dụ của MTA.

Đại diện của người sử dụng thư tín: chương trình thư điện tử được sử dụng để xây dựng thông điệp thư điện tử và truyền đạt với MTA để gửi/ truy cập các thông điệp thư tín. Microsoft Outlook là một ví dụ điển hình của MUA.

MTA's thường được phục vụ như “các máy chủ truy cập thư tín” và điển hình có thể phục vụ hàng trăm hoặc nhiều hơn nữa MUA's.

MUA's chịu trách nhiệm cho việc xây dựng các thông điệp thư điện tử phù hợp với Internet Electronic Mail Specifications (quy định kỹ thuật thư điện tử internet) được định danh ở trên.

Phần này mô tả sự nối kết của một thông điệp phụ thuộc *ebXML* để truyền tải thông qua thư điện tử (eMail) từ hình phối cảnh của MUA. Không có sự nỗ lực nào được tạo ra để trao đổi sự kết nối của thông điệp *ebXML* lên trên SMTP từ lập trường của MTA.

B.3.1 Mức tối thiểu của các giao thức hỗ trợ

Simple Mail Transfer Protocol [RFC2821] : Giao thức chuyển giao thư điện tử đơn [RFC2821]
MIME [RFC2045] và [RFC2046]

Multipart/Related MIME [RFC2387] : đa vai trò/có liên quan MIME [RFC2387]

B.3.2 Việc gửi các thông điệp ebXML lên SMTP

Trước khi gửi các thông điệp lên SMTP, một thông điệp ebXML PHẢI được định dạng theo *dịch vụ thông điệp ebXML Specification* (quy định dịch vụ thông điệp ebXML). Hơn nữa, các thông điệp cũng PHẢI phù hợp với cú pháp, định dạng và quy tắc mã hoá lý thuyết bởi MIME [RFC2045], [RFC2046] và [RFC2387]. Nhiều kiểu dữ liệu mà một phần có lẽ yêu cầu truyền tải thông qua thư điện tử được trình bày giống như ký tự 8 bit hoặc dữ liệu nhị phân. Quả thật, dữ liệu không thể được truyền tải lên SMTP [RFC 2821] hạn chế thông điệp thư điện tử với dữ liệu 7 bit US – ASCII với các dòng không dài hơn 1000 ký tự kể cả bất kỳ máy tách dòng CRLF nào. Nếu Dịch vụ thông điệp Handle (Trình quản lý dịch vụ thông điệp)(MSH) gửi định danh MTA nhận hoặc bất cứ MTA's trung gian nào bị giới hạn để quản lý dữ liệu 7 bit thì bất kỳ phần tài liệu nào sử dụng sự trình bày 8 bit (hoặc nhị phân) PHẢI được “biến đổi” theo các quy tắc mã hoá trên danh nghĩa trong phần 6 của MIME [RFC 2045]. Trong trường hợp tại nơi Trình quản lý dịch vụ thông điệp (MSH) biết rằng MTA nhận (receiving MTA) và tất cả MTA's trung gian có khả năng trong việc quản lý dữ liệu 8 bit thì sự không biến đổi là cần thiết trên bất cứ phần nào của thông điệp ebXML.

Các quy tắc định dạng để truyền tải thông điệp ebXML qua SMTP như sau:

- nếu việc sử dụng SMTP [RFC 2821] giới hạn truyền tải các đường dẫn, áp dụng việc chuyển giao mã hoá tới tất cả dữ liệu 8 bit thì PHẢI được truyền tải trong thông điệp ebXML, theo các quy tắc mã hoá đã xác định trong phần 6 của MIME [RFC 2045]. Tiêu đề nội dung – Chuyển giao – Mã hoá (Content – Transfer – Encoding – MIME PHẢI được chứa trong phần bên ngoài MIME của bất kỳ phần thân nào đã được chuyển giao (mã hoá);
- content – Type: tiêu đề Multipart/Related (Đa phần/ liên quan) MIME với các tham số kết hợp từ thông điệp ebXML Envelope (phong bì thông điệp ebXML) PHẢI xuất hiện giống như tiêu đề eMail MIME;
- tất cả tiêu đề MIME khác tạo thành thông điệp ebXML Envelope cũng PHẢI trở thành một phần của tiêu đề eMail MIME;
- trường phần tử **Header** SOAP Action MIME cũng PHẢI được bao gồm trong tiêu đề eMail MIME và có thể có giá trị của ebXML:

SOAPAction: “**ebXML**”

- tiêu đề “MIME – Version(phiên bản): 1.0” PHẢI xuất hiện giống như tiêu đề eMail MIME;
- tiêu đề eMail “To” PHẢI bao gồm SMTP [RFC 2821] phục tùng mệnh lệnh địa chỉ eMail của các người gửi ebXML Trình quản lý dịch vụ thông điệp (Trình quản lý dịch vụ thông điệp);

TCVN ISO/TS 15000-2 : 2007

- tiêu đề eMail “From”: PHẢI chứa SMTP [RFC2821] phục tùng mệnh lệnh địa chỉ eMAIL của các người gửi ebXML Trình quản lý dịch vụ thông điệp (Trình quản lý dịch vụ thông điệp);
- tạo “Date”: tiêu đề eMail phù hợp với SMTP [RFC2821];
- các phần tử khác có thể xuất hiện trong tiêu đề thông điệp eMail phù hợp với SMTP [RFC2821] và MIME [RFC 2045], tuy nhiên ebXML Trình quản lý dịch vụ các thông điệp có thể chọn cách bỏ qua chúng.

Ví dụ dưới đây cho thấy một ví dụ tối thiểu của thông điệp eMail chứa thông điệp ebXML:

```
From: ebXMLhandler@example.com
To: ebXMLhandler@example2.com
Date: Thu, 08 Feb 2001 19:32:11 CST
MIME-Version: 1.0
SOAPAction: "ebXML"
Content-type: multipart/related; boundary="Boundary"; type="text/xml";
             start="<ebxhheader111@example.com>"
```

This is an ebXML SMTP Example

```
--Boundary
Content-ID: <ebxhheader111@example.com>
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
  xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
    http://www.oasis-open.org/committees/ebxml-
      msg/schema/envelope.xsd">
<SOAP:Header xmlns:eb="http://www.oasis-open.org/committees/ebxml-
  msg/schema/msg-header-2_0.xsd"
  xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-
    msg/schema/msg-header-2_0.xsd
      http://www.oasis-open.org/committees/ebxml-
        msg/schema/msg-header-2_0.xsd">
<eb:MessageHeader SOAP:mustUnderstand="1" eb:version="2.0">
  <eb:From>
    <eb:PartyId>urn:duns:123456789</eb:PartyId>
  </eb:From>
  <eb:To>
    <eb:PartyId>urn:duns:912345678</eb:PartyId>
  </eb:To>
  <eb:CPAId>20001209-133003-28572</eb:CPAId>
  <eb:ConversationId>20001209-133003-28572</eb:ConversationId>
  <eb:Service>urn:services:SupplierOrderProcessing</eb:Service>
  <eb:Action>NewOrder</eb:Action>
  <eb:MessageData>
    <eb:MessageId>20001209-133003-28572@example.com</eb:MessageId>
    <eb:Timestamp>2001-02-15T11:12:12</eb:Timestamp>
  </eb:MessageData>
  <eb:DuplicateElimination/>
</eb:MessageHeader>
</SOAP:Header>
<SOAP:Body xmlns:eb="http://www.oasis-open.org/committees/ebxml-
  msg/schema/msg-header-2_0.xsd"
  xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-
    msg/schema/msg-header-2_0.xsd
      http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
        header-2_0.xsd">
```



```

<eb:Manifest eb:version="2.0">
  <eb:Reference xlink:href="cid:ebxmlpayload111@example.com"
    xlink:role="XLinkRole"
    xlink:type="simple">
    <eb:Description xml:lang="en-US">Purchase Order 1</eb:Description>
  </eb:Reference>
</eb:Manifest>
</SOAP:Body>
</SOAP:Envelope>

--Boundary
Content-ID: <ebxmhheader111@example.com>
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>
<purchase_order>
  <po_number>1</po_number>
  <part_number>123</part_number>
  <price currency="USD">500.00</price>
</purchase_order>

--Boundary--

```

B.3.3 Thông điệp phản hồi

Tất cả các thông điệp phản hồi, bao gồm các lỗi và các thông báo nhận được tạo ra *không đồng bộ* trong ebXML Trình quản lý dịch vụ các thông điệp (Trình quản lý dịch vụ thông điệp). Mỗi thông điệp phản hồi PHẢI được tạo nên phù hợp với các quy tắc lý thuyết trong phần B.3.2

Tất cả ebXML Trình quản lý dịch vụ thông điệp (Trình quản lý dịch vụ thông điệp) PHẢI có khả năng nhận thông điệp thông báo lỗi tạo ra gửi bằng MTA. Một MSH nhận được thông điệp thông báo lỗi tạo ra nên xem xét thông điệp để quyết định thông điệp ebXML nào, gửi bằng MSH, dẫn đến thông điệp tạo ra lỗi. MSH nên cố gắng định danh ứng dụng chịu trách nhiệm gửi các thông điệp khó chịu do lỗi thất bại. MSH nên cố gắng thông báo cho ứng dụng rằng thông điệp tạo ra lỗi đã tìm thấy. Nếu MSH không thể xác định nguồn của thông điệp gây khó chịu thì quản trị viên MSH nên được thông báo.

Nếu MSH's không thể định danh thông điệp nhận giống như thông điệp ebXML hợp lệ hoặc thông điệp tạo ra lỗi nên giữ lại thông điệp chưa định danh trong thư mục "thư không ai nhận".

MSH nên đặt mục nhập trong bản ghi kiểm định để cho biết sự sắp xếp mỗi thông điệp được nhận.

B.3.4 Điều khiển truy cập

Các phương tiện có thể bảo vệ các bộ quản lý dịch vụ thông điệp ebXML (ebXML Trình quản lý dịch vụ các thông điệp) của chúng từ việc truy cập trái phép xuyên suốt việc sử dụng cơ chế điều khiển truy cập. Quá trình thẩm định truy cập SMTP đã mô tả trong "SMTP **Service** Extension for Authentication" (dịch vụ mở rộng thẩm định SMTP) [RFC 2554] định rõ cơ chế điều khiển truy cập yêu cầu ebXML để bảo vệ Trình quản lý dịch vụ thông điệp ebXML cơ bản SMTP từ truy cập trái phép.

B.3.5 Độ tin cậy và mức an ninh giao thức truyền tải

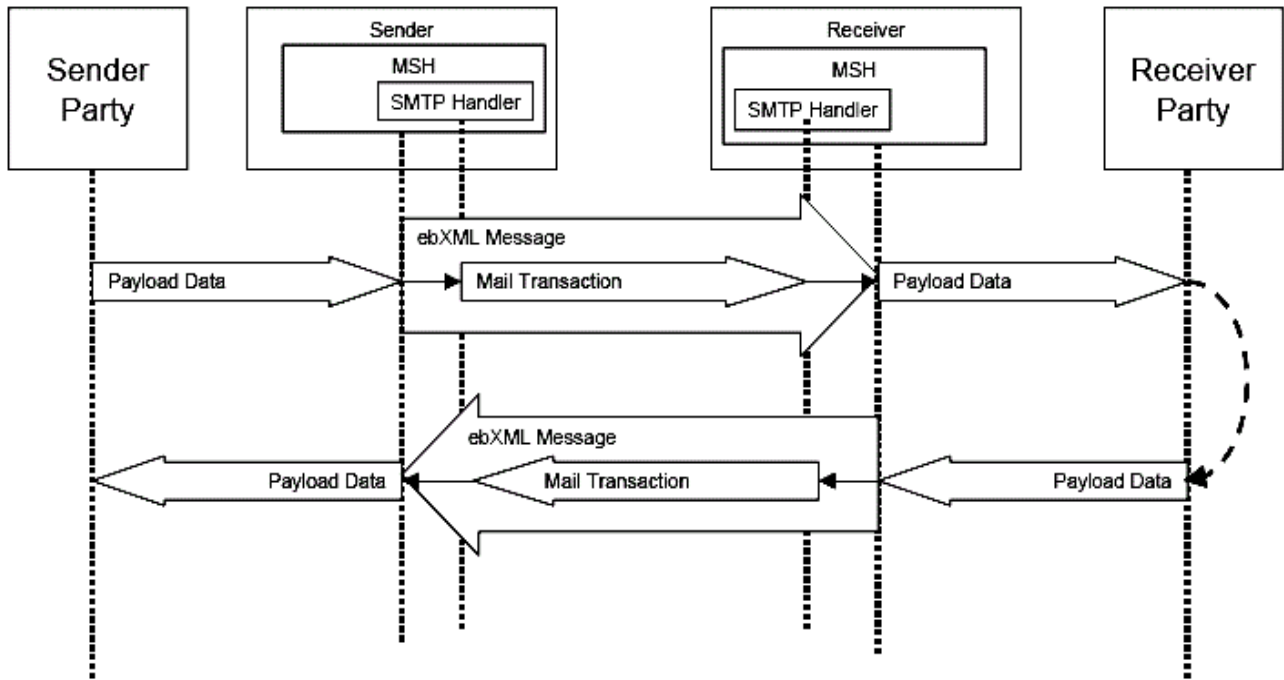
Trình quản lý dịch vụ thông điệp ebXML (ebXML Trình quản lý dịch vụ thông điệp) có thể sử dụng sự mật mã hóa lớp truyền tải để bảo vệ sự cẩn mật của thông điệp ebXML. Đặc điểm kỹ thuật (Đặc tả)

TCVN ISO/TS 15000-2 : 2007

IETF “SMTP **Service** Extension for Secure SMTP over TLS” [RFC2487] cung cấp các chi tiết kỹ thuật đặc trưng và danh sách các tùy chọn được quy định có thể được sử dụng.

B.3.6 Mô hình SMTP

Tất cả các thông điệp *dịch vụ thông điệp ebXML* mang lại giống như thư tín trong sự giao dịch thư tín SMTP [RFC2821] (SMTP Mail Transaction) thể hiện trong hình B1



Hình B-1 – Mô tả thư tín SMTP

B.4 Lỗi truyền thông trong thông điệp xác thực

Nếu người gửi hoặc người nhận phát hiện ra một cấp độ lỗi trong giao thức truyền thông (giống như HTTP, SMTP hoặc lỗi FTP) và việc truyền thông điệp tin cậy (thông điệp xác thực) được sử dụng thì bộ quản lý phục hồi truyền tải thích hợp **PHẢI** thực hiện các trình tự phục hồi. Nếu chỉ lỗi không có khả năng phục hồi thì Việc truyền thông điệp tin cậy 2735 phục hồi lại được vị trí (xem phần 6).

Phụ lục C

Hỗ trợ các dịch vụ an ninh

Cấu trúc chung của Quy định dịch vụ thông điệp ebXML (*dịch vụ thông điệp ebXML Specification*) được dùng để hỗ trợ cho tất cả các dịch vụ an ninh đòi hỏi giao dịch điện tử. Bảng sau đây kết hợp với các dịch vụ an ninh *Trình quản lý dịch vụ thông điệp* vào trong việc thiết lập hiện trạng an ninh. Các hiện trạng này hoặc sự kết hợp của các hiện trạng này, hỗ trợ chính sách an ninh riêng biệt của đa số các người sử dụng ebXML. Đúng với trạng thái chưa chín muồi của các đặc điểm kỹ thuật an ninh XML, phiên bản này của quy định kỹ thuật đòi hỏi hỗ trợ cho chỉ hiện trạng 0 hoặc 1.

Việc này không ngăn ngừa các người sử dụng đối với các tính năng an ninh bổ xung để bảo vệ sự trao đổi ebXML, tuy nhiên khả năng thao tác giữa các phần sử dụng bất cứ hiện trạng nào khác hơn 0 và 1 không thể được bảo đảm .

Xuất hiện trong đường cơ sở		Chữ ký số lưu lâu dài	Sự thẩm định quyền ngắn hạn	Việc nhận được ký lâu dài	Tình trạng toàn vẹn ngắn hạn	Sự cần mật lâu dài	Sự cần mật ngắn hạn	Quyền hạn lâu dài	Quyền hạn ngắn hạn	Loại thời gian tín nhiệm	Mô tả Hồ sơ
√	Hồ sơ 0										Các dịch vụ không an ninh được áp dụng cho dữ liệu
√	Hồ sơ 1	√									<i>MSH gửi</i> áp dụng các cấu trúc XML/DSIG với thông điệp
	Hồ sơ 2		√						√		<i>MSH gửi</i> xác thực và <i>MSH nhận</i> cho phép người gửi dựa trên các tài liệu uỷ nhiệm kênh truyền thông
	Hồ sơ 3		√				√				<i>MSH gửi</i> xác thực và cả <i>MSH</i> dàn xếp kênh an ninh để truyền tải dữ liệu.
	Hồ sơ 4		√		√						<i>MSH gửi</i> xác thực, <i>MSH nhận</i> thực hiện tình trạng nguyên vẹn kiểm tra việc sử dụng giao thức liên lạc.
	Hồ sơ 5		√								<i>MSH gửi</i> chỉ xác thực kênh truyền thông (ví dụ SSL 3.0 trên TCP/IP)
	Hồ sơ 6	√					√				<i>MSH gửi</i> áp dụng các cấu trúc XML/DSIG đối với thông điệp và bỏ qua trong kênh phương tiện liên lạc an ninh.
	Hồ sơ 7	√		√							<i>MSH gửi</i> áp dụng các cấu trúc XML/DSIG đối

										với thông điệp và <i>MSH nhận</i> gửi lại một chữ ký đã nhận được
Hồ sơ 8	√		√			√				Là sự kết hợp của Mô tả 6 và 7
Hồ sơ 9	√								√	Mô tả 5 với loại thời gian tin cậy đã áp dụng
Hồ sơ 10	√		√						√	Mô tả 9 với <i>MSH nhận</i> gửi lại một chữ ký nhận được.
Hồ sơ 11	√					√			√	Mô tả 6 với <i>MSH nhận</i> áp dụng loại thời gian tin cậy
Hồ sơ 12	√		√			√			√	Mô tả 8 với <i>MSH nhận</i> áp dụng loại thời gian tin cậy
Hồ sơ 13	√				√					<i>MSH gửi</i> áp dụng các cấu trúc XML/DSIG đối với thông điệp và áp dụng cần mật các cấu trúc (XMLEncryption)
Hồ sơ 14	√		√		√					Mô tả 13 với xác nhận ký hiệu
Hồ sơ 15	√		√						√	<i>MSH gửi</i> áp dụng các cấu trúc XML/DSIG đối với thông điệp, loại thời gian tin nhiệm được thêm vào thông điệp, <i>MSH nhận</i> trả lại một sự xác nhận ký hiệu
Hồ sơ 16	√				√				√	Mô tả 13 với loại thời gian tin cậy đã áp dụng
Hồ sơ 17	√		√		√				√	Mô tả 14 với loại thời gian tin cậy đã áp dụng
Hồ sơ 18	√						√			<i>MSH gửi</i> áp dụng các cấu trúc XML/DSIG đối với thông điệp và đẩy mạnh khả năng quyền hạn [SAML]
Hồ sơ 19	√		√				√			Mô tả 18 với <i>MSH nhận</i> điều hướng lại xác nhận ký hiệu
Hồ sơ 20	√		√				√		√	Mô tả 19 với loại thời gian tin cậy được áp dụng đối với thông điệp <i>MSH gửi</i>
Hồ sơ 21	√		√		√		√		√	Mô tả 19 với <i>MSH gửi</i> áp dụng cho các cấu trúc tin cậy(XML-Encryption)
Hồ sơ 22					√					<i>MSH gửi</i> tóm lược thông điệp trong cấu trúc tin cậy (XML-Encryption)

Tài liệu tham khảo

(quy định)

[RFC2119] Key Words for use in RFCs to Indicate Requirement Levels, Internet Engineering Task Force, March 1997: (Các Từ khoá sử dụng trong RFCs để cho biết các cấp độ yêu cầu, thao tác kỹ thuật Internet Force, tháng 3/1997)

[RFC2045] Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Thông điệp Bodies, N Freed & N Borenstein, Published November 1996 ([RFC2045] Đuôi mở rộng thư điện tử vạn năng (MIME) phần I: Định dạng phần chính thư điện tử 2751, N Freed & N Borenstein, Published tháng 11/1996)

[RFC2046] Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. N. Freed, N. Borenstein. November 1996. (Đuôi mở rộng thư điện tử (MIME) phần II: Các dạng truyền thông. N Freed, N.)

[RFC2246] Dierks, T. và C. Allen, "The TLS Protocol", January 1999.: Dierks, T. và C.Allen, ("Giao thức TLS", tháng 1/1999).

[RFC2387] The MIME Multipart/Related Content-type. E. Levinson. August 1998.(Dạng nội dung MIME Multipart/Related. E. Levinson. Tháng 8/1998)

[RFC2392] Content-ID và Thông điệp-ID Uniform Resource Locators. E. Levinson, August 1998 (Các ranh giới nguồn đồng bộ Nội dung – ID và Thông điệp ID. E.Levinson. Tháng 8/1998 2757.)

[RFC2396] Uniform Resource Identifiers (URI): Generic Syntax. T Berners-Lee, August 1998 (Định danh nguồn thống nhất (URI): Cú pháp chung. T.Berners-Lee, tháng 8/1998)

[RFC2402] IP Authentication Header. S. Kent, R. Atkinson. November 1998. RFC2406 IP (Sự thẩm định quyền tiêu đề IP. S.Kent, R.Atkinson. Tháng 11/1988. RFC2406 IP)

Encapsulating Security Payload (ESP). S. Kent, R. Atkinson. November 1998. (Vùng mang thông tin an ninh tóm lược (ESP). S.Kent, R. Atkinson. November 1998.)

[RFC2487] SMTP Service Extension for Secure SMTP over TLS. P. Hoffman, January 1999. 2761 (Đuôi mở rộng dịch vụ SMTP để bảo đảm SMTP lên TLS. P.Hoffman, tháng 1/1999)

[RFC2554] SMTP Service Extension for Authentication. J. Myers. March 1999. (Đuôi mở rộng dịch vụ SMTP cho việc thẩm định quyền. J.Myers. tháng 3/1999.)

[RFC2821] Simple Mail Transfer Protocol, J. Klensin, Editor, April 2001 Obsoletes RFC 821 2763 (Giao thức chuyển giao thư tin đơn, J.Klensin, Editor, tháng 4/2001 Quá hạn RFC 821)

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. và T. Berners-Lee,

TCVN ISO/TS 15000-2 : 2007

"Hypertext Transfer Protocol, HTTP/1.1", June 1999. 2765: (*Giao thức chuyển giao siêu văn bản, HTTP/1.1* Tháng 6/1999)

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., Sink, E. và L. Stewart, "HTTP Authentication: Basic và Digest Access Authentication" (*Sự thẩm định quyền HTTP: Sự thẩm định quyền truy cập tóm tắt và cơ bản*), tháng 6/1999

[RFC2817] Khare, R. và S. Lawrence, "Upgrading to TLS Within HTTP/1.1" (*Việc nâng cấp TLS trong HTTP/1.1*) tháng 5/2000

[RFC2818] Rescorla, E., "HTTP Over TLS", (HTTP trên TLS) tháng 5/2000 [SOAP] Giao thức truy cập đối tượng đơn

[SOAP] W3C-Draft-Simple Object Access Protocol (SOAP) (*Giao thức truy cập đối tượng đơn– Nháp – W3C*) v1.1, Don Box, DevelopMentor; David Ehnebuske, IBM; Gopal Kakivaya, Andrew Layman, Henrik Frystyk Nielsen, Satish Thatte, Microsoft; Noah Mendelsohn, Lotus Development Corp.; Dave Winer, UserLand Software, Inc.; W3C Note 08 May 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

[SOAPAttach] SOAP Các thông điệp with Attachments (*Các thông điệp SOAP với phần đính kèm*), John J. Barton, Hewlett Packard Labs; Satish Thatte và Henrik Frystyk Nielsen, Microsoft, Published Oct 09 2000 <http://www.w3.org/TR/2000/NOTE-SOAP-attachments-20001211>

[SSL3] A. Frier, P. Karlton và P. Kocher, "The SSL 3.0 Protocol" (*Giao thức SSL 3.0*), Netscape Communications Corp., Nov 18, 1996.

[UTF-8] UTF-8 là mã hoá phù hợp với ISO/IEC 10646. Xem [XML] về việc sử dụng các quy ước.

[XLINK] W3C XML Linking Recommendation (*Giới thiệu kết nối XML W3C*), <http://www.w3.org/TR/2001/REC-xlink-20010627/>

[XML] W3C Recommendation: Extensible Markup Language (XML) 1.0 (Second Edition), 2783 : [XML] (*Giới thiệu W3C: Ngôn ngữ đánh dấu mở rộng (XML) 1.0 (ấn bản thứ 2) tháng 10/2000*) <http://www.w3.org/TR/2000/REC-xml-20001006>

[XMLC14N] W3C Recommendation Canonical XML 1.0.; [XMLNS] (*Giới thiệu W3C đạt tiêu chuẩn XML 1.0*) <http://www.w3.org/TR/2001/REC-xml-c14n-20010315>

[XMLNS] W3C Recommendation for Namespaces in XML, World Wide Web Consortium, 14 2787: [XMLNS] (*Sự giới thiệu W3C cho Namespaces trong XML, World Wide Web Consortium, 14 tháng 11/1999*)

<http://www.w3.org/TR/1999/REC-xml-names-19990114/>

[XMLDSIG] Joint W3C/IETF XML-Signature Syntax và Processing specification, (*Đầu nối cú pháp chữ ký W3C/IETF XML và đặc điểm kỹ thuật xử lý*) <http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/>.

[XMLMedia] [RFC 3023](#), XML Media Types. M. Murata, S. St. Laurent, January (*các dạng truyền thông XML. M.Mutura, S.St. Laurent, tháng 1/2001*)

[XPointer] XML Pointer Language (XPointer) Version 1.0, W3C Candidate Recommendation 11
September 2001, (*ngôn ngữ con trỏ XML*) <http://www.w3.org/TR/2001/CR-xptr-20010911/>.

Tài liệu tham khảo

(tham khảo)

[ebCPP] ebXML Collaboration Protocol Profile và Agreement specification, Version 1.0, (*[ebCPP] Hiện trạng giao thức cộng tác ebXML và quy định kỹ thuật thoả thuận, Phiên bản 1.0 xuất bản 10/5/2001*) <http://www.ebxml.org/specs/ebCCP.doc>

[ebBPSS] ebXML Business Process Specification Schema, version 1.0, published 27 April 2001 (*Giản đồ quy định quá trình giao dịch ebXML, phiên bản 1.0, xuất bản 27/4/2001*) <http://www.ebxml.org/specs/ebBPSS.pdf>.

[ebTA] ebXML Technical Architecture, version 1.04 published 16 February, 2001 (*Cấu trúc kỹ thuật ebXML, phiên bản 1.04 xuất bản ngày 16/2/2001*) <http://www.ebxml.org/specs/ebTA.doc>

[ebRS] ebXML Registry Services Specification, version 2.0, published 6 December 2001 (*Đặc tả kỹ thuật phục vụ đăng ký ebXML, phiên bản 2.0, công bố ngày 6/12/2001*) <http://www.oasis-open.org/committees/regrep/documents/2.0/specs/ebrs.pdf>, xuất bản ngày 5/12/2001. <http://www.oasis-open.org/committees/regrep/documents/2.0/specs/ebrim.pdf>

[ebREQ] ebXML Requirements Specification (*Đặc tả yêu cầu ebXML*) <http://www.ebxml.org/specs/ebREQ.pdf>, công bố ngày 8/5/2001.

[ebGLOSS] ebXML Glossary (*Bảng chú giải thuật ngữ ebXML*) <http://www.ebxml.org/specs/ebGLOSS.doc>, công bố ngày 11/5/2001.

[PGP/MIME] [RFC2015](#), "MIME Security with Pretty Good Privacy (PGP)", M. Elkins. October 1996.

[SAML] Security Assertion Markup Language (*Ngôn ngữ xác nhận an ninh*) <http://www.oasis-open.org/committees/security/docs/draft-sstc-use-strawman-03.html>

[S/MIME] [RFC 2311](#), "S/MIME Version 2 Thông điệp Specification" (*Đặc tả thông điệp phiên bản 2 S/MIME*) S. Dusse, P. Hoffman, B.

Ramsdell, L. Lundblade, L. Repka. March 1998.

[S/MIMECH] [RFC 2312](#), "S/MIME Version 2 Certificate Handling" (*Giấy Chứng nhận kênh quản lý phiên bản 2 S/MIME*), S. Dusse, P. Hoffman, B. Ramsdell, J. Weinstein. tháng 3/1998.

[S/MIMEV3] [RFC 2633](#) S/MIME Version 3 Message Specification (*Đặc tả thông điệp phiên bản 3 S/MIME RFC 2633*). B. Ramsdell, Ed June 1999.

[secRISK] ebXML Technical Architecture Risk Assessment Technical Report, (*Báo cáo đánh giá kỹ thuật về độ rủi ro cấu trúc kỹ thuật ebXML phiên bản 0.36 2817 công bố 20/4/2001*)

[XMLSchema] W3C XML Schema Recommendation (*Giới thiệu giản đồ XML W3C*)

<http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/>

<http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>

<http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>

[XMTP] XMTP - Extensible Mail Transport Protocol (*Giao thức truyền tải thư tín mở rộng*)

<http://www.openhealth.org/documents/xmtp.htm>
