

TTA Standard

정보통신단체표준(국문표준)

TTAS.IT-X1141_1

개정일: 2006 년 12 월 27 일

SAML 2.0 주장과 프로토콜

SAML 2.0 Assertions and Protocols

SAML 2.0 주장과 프로토콜

SAML 2.0 Assertions and Protocols



본 문서에 대한 저작권은 TTA에 있으며, TTA와 사전 협의 없이 이 문서의 전체 또는 일부를 상업적 목적으로 복제 또는 배포해서는 안 됩니다.

Copyright© Telecommunications Technology Associations 2006. All Rights Reserved.

서 문

1. 표준의 목적

SAML(Security Assertion Markup Language) 2.0 은 분산된 환경에서 인증, 인가 및 속성 정보를 교환하기 위한 XML-기반 프레임워크이다. 이름에서 나타나듯이, SAML 은 비즈니스 엔티티들이 어떤 주체의 신원, 속성 그리고 권한부여에 대한 주장을 파트너 회사 또는 다른 엔터프라이즈 응용 등과 같은 다른 엔티티들에게 보장할 수 있도록 해 준다. 이 표준은 SAML 주장들의 구조와 관련된 프로토콜의 구조를 정의하고 또한 SAML 시스템을 관리하는데 관련된 처리 규칙들을 정의한다.

이 표준은 ITU-T X.1141 “Security Assertion Markup Language (SAML 2.0)”을 근거로 한 국내 표준으로 원문의 다음 내용을 포함하고 있다.

- CL 1. 범위
- CL 2. 참고문헌
- CL 3. 용어정의
- CL 4. 약어
- CL 5. 관례
- CL 7. 공통 데이터 타입
- CL 8. SAML 주장과 프로토콜

2. 주요 내용 요약

이 표준은 주장과 프로토콜에 대한 문법 및 의미를 정의하고 또한 이것들과 관련된 처리 규칙을 정의한다. 이 표준은 SAML 버전 관리와 SAML 에서 사용되는 XML 서명과 암호화의 문법에 대하여 기술한다. 확장성 및 SAML 에서 정의된 식별자들이 또한 기술된다.

3. 표준 적용 산업 분야 및 산업에 미치는 영향

본 표준은 웹 싱글사인온, 속성 정보 기반 인가와 웹 서비스 보호에서 사용될 수 있다. 따라서, 본 표준은 ID 관리 분야와 웹 서비스 정보보호 분야에 직접적으로 적용되며, 정보보호 산업의 핵심 요소로 활용될 수 있다. 또한, ID 연계의 핵심 기술을 제공함으로써, 기업간 협업을 용이하게 함으로써 새로운 서비스를 창출하고 시장을 활성화할 수 있다.

4. 참조 표준(권고)

4.1. 국외 표준(권고)

- ITU-T X.1141, 'Security Assertion Markup Language (SAML 2.0)', 2006.06.

4.2. 국내 표준

- TTAS.OT-10.0042, 'SAML 구문과 프로토콜', 2005.12.

5. 참조 표준(권고)과의 비교

5.1. 참조 표준(권고)과의 관련성

본 ITU-T X.1141, 'Security Assertion Markup Language (SAML 2.0)'을 근거로 한 국내표준임. TTAS.OT-10.0042 는 OASIS 의 SAMLv1.0 을 기준으로 개발된 표준이었으나 현재 SAMLv2.0 표준이 ITU-T 표준으로 개발되어 이를 개정함.

5.2. 참조한 표준(권고)과 본 표준의 비교표

상기 국제 권고에 대한 추가사항은 없으며, 장 구성은 다음과 같음.

ITU-T X.1141	본 표준	비고
1. 범위	1.1. 범위	동일(번역)
2. 참고문헌	1.2. 참고문헌	동일(번역)
3. 용어정의	1.3. 용어정의	동일(번역)
4. 약어	1.4. 약어	동일(번역)
5. 관례	1.5. 관례	동일(번역)
7. 공통 데이터 타입	1.6. 공통 데이터 타입	동일(번역)
8. SAML 주장과 프로토콜	2~8. SAML 주장과 프로토콜	동일(번역)

6. 지식 재산권 관련 사항

본 표준의 '지식 재산권 요약서' 제출 현황은 TTA 웹사이트에서 확인할 수 있다.

※본 표준을 이용하는 자는 이용함에 있어 지식 재산권이 포함되어 있을 수 있으므로, 확인 후 이용한다.

※본 표준과 관련하여 접수된 요약서 이외에도 지식 재산권이 존재할 수 있다.

7. 시험 인증 관련 사항

7.1. 시험 인증 대상 여부

- 해당 사항 없음.

7.2. 시험 표준 제정 현황

- 해당 사항 없음.

8. 표준의 이력 정보

8.1. 표준의 이력

판수	제정·개정일	제정·개정 내역
제 1 판	2005. 12. 21.	제정 TTAS.IT-X1141
제 2 판	2006. 12. 27.	개정 TTAS.IT-X1141_1

8.2. 주요 개정 사항


- 해당 사항 없음.

Preface

1. Purpose of Standard

SAML(Security Assertion Markup Language) is an XML-based framework for communicating user authentication, entitlement, and attribute information among disparate Web access management and security products. As its name suggests, SAML allows business entities to make assertions regarding the identity, attributes, and entitlements of a subject (an entity that is often a human user) to other entities, such as a partner company or another enterprise application. This standard defines both the structure of SAML assertions, and an associated set of protocols, in addition to the processing rules involved in managing a SAML system.

This standard is a domestic standard based on ITU-T X.1141 “Security Assertion Markup Language (SAML 2.0)” and contains the following contents of the original standard.

- 
- CL 1. Scope
 - CL 2. References
 - CL 3. Definitions
 - CL 4. Abbreviations
 - CL 5. Conventions
 - CL 7. Common data types
 - CL 8. SAML assertions and protocols

2. Summary of Contents

The standard defines the syntax and semantics for assertions and protocols. It defines also the processing rules related to assertions and protocols. Then, it specifies SAML versioning and XML Signature and Encryption syntax and processing which is used in SAML. Extensibility and SAML-defined identifiers also specified in this specification.

3. Applicable Fields of Industry and its Effect

This standard can be used in as web single-sign on, attribute information based authorization and web service security. Therefore, it is directly applicable to security areas such as ID Management and web service security. It is also applicable to other information security industry as essential component. In addition, it provides essential technology for ID federation, which makes companies' collaboration easy and so creates new service and revitalize IT market.

4. Reference Standards(Recommendations)

4.1. International Standards(Recommendations)

- ITU-T X.1141, "Security Assertion Markup Language (SAML 2.0)", 2006.06.

4.2. Domestic Standards

- TTAS.OT-10.0042, "SAML Assertions and Protocol", 2005.12.

5. Relationship to Reference Standards(Recommendations)

5.1. Relationship of Reference Standards(Recommendations)

This standard is a domestic standard based on ITU-T X.1141, "Security Assertion Markup Language (SAML 2.0)". This standard updated the TTAS.OT-10.0042 based on OASIS standard as SAMLv1.0.

5.2. Differences between Reference Standard(Recommendation) and this Standard

This standard has no additional contents as to the international recommendations. The differences between the recommendation and this standard are as follows.

ITU-T X.1141	This Standard	
1. Scope	1.1. Scope	equaled(translated)
2. References	1.2. References	equaled(translated)
3. Definitions	1.3. Definitions	equaled(translated)
4. Abbreviations	1.4. Abbreviations	equaled(translated)
5. Conventions	1.5. Conventions	equaled(translated)
7. Common data types	1.6. Common data types	equaled(translated)
8. SAML assertions and protocols	2~8. SAML assertions and protocols	equaled(translated)

6. Statement of Intellectual Property Rights

IPRs related to the present document may have been declared to TTA. The information pertaining to these IPRs, if any, is available on the TTA Website.

No guarantee can be given as to the existence of other IPRs not referenced on the TTA website.

And, please make sure to check before applying the standard.

7. Statement of Testing and Certification

7.1. Object of Testing and Certification

– None

7.2. Standards of Testing and Certification

– None

8. History of Standard

8.1. Change History

Edition	Issued date	Outline
The 1st edition	2005. 12. 21.	Established TTAS.IT-X1141
The 2nd edition	2006. 12. 27.	Revised TTAS.IT-X1141_1

8.2. Revisions

– None



목 차

1. SAML 2.0 개요	1
2. SAML 주장	24
3. SAML 프로토콜	62
4. SAML 버전	109
5. SAML 과 XML 서명 문법 및 처리	113
6. SAML 과 XML 암호화 문법 및 처리	124
7. SAML 확장성	125
8. SAML 에서 정의된 식별자	128

Contents

1. SAML 2.0 Introduction	1
2. SAML Assertions	24
3. SAML Protocols	62
4. SAML Versioning	109
5. SAML and XML Signature Syntax and Processing	113
6. SAML and XML Encryption Syntax and Processing	124
7. SAML Extensibility	125
8. SAML-Defined Identifiers	128

SAML 2.0 주장과 프로토콜

(SAML 2.0 Assertions and Protocols)

1. SAML 2.0 개요

1.1. 범위(Scope)

SAML 2.0 은 시스템 엔티티가 어떤 주체에 대하여 생성한 주장의 문법과 처리 규칙을 정의한다. 이와 같은 주장을 만들거나 또는 의지하기 위해, SAML 시스템 엔티티들은 주장 자체 또는 주장의 주체에 대한 내용을 통신하기 위해 다른 프로토콜을 사용할 수 있다. SAML 2.0 은 SAML 보장의 구조, 관련된 프로토콜 집합, 그리고 SAML 시스템을 관리하는데 관련된 처리 규칙들을 정의한다.

SAML 주장과 프로토콜 메시지들은 XML 로 인코딩되어 있으며, XML 네임스페이스를 사용한다. 이것들은 일반적으로 HTTP POST 또는 XML 로 인코딩된 SOAP 메시지와 같은 전송을 위한 다른 구조에 내장된다. SAML 2.0 은 또한 SAML 프로토콜 메시지들을 내장하고 전송하기 위한 프레임워크를 제공하는 SAML 바인딩을 명기한다. 더욱이, SAML 2.0 은 SAML 특징들을 사용할 때, 특정 사용예(use case)를 달성하고 상호운용성을 달성하기 위해, SAML 주장과 프로토콜을 어떻게 사용해야 하는지에 대한 기본 프로파일 집합을 제공한다.

SAML 2.0 은 다음을 정의한다.

- SAML 에 대한 적합성 요구사항;
- SAML 주장과 프로토콜:
 - SAML 주장 스키마,
 - SAML 프로토콜 스키마;
- SAML 바인딩;
- SAML 프로파일:
 - SAML ECP 프로파일 스키마,
 - SAML X.500/LDAP 속성 프로파일 스키마,
 - SAML DCE PAC 속성 프로파일 스키마,
 - SAML XACML 속성 프로파일 스키마;
- SAML 메타데이터;
- SAML 메타데이터 스키마;
- SAML 인증 문맥.

1.2. 참고문헌

다음 권고안들과 다른 참조들은 SAML 2.0 에서 참조되는 것들이다. SAML 2.0 의 발간시에는 모두 유효한 상태이다. 모든 권고안들과 다른 참조들은 개정될 수 있으며, SAML 2.0 에 기반으로 하는 모든 사용자들은 아래 나열된 권고안들과 다른 참조들에 대하여 가장 최신 판을 적용할 수 있다. ITU 의 전기통신 표준국(Telecommunications Standardization Bureau)에서 현재 유효한 ITU-T 권고안들의 리스트를 유지한다. IETF 는 최근에 폐지된 것들과 함께 RFC 리스트를 유지한다. W3C, Unicode Consortium 과 Liberty Alliance 도 가장 최신의 권고안들과 다른 문서들에 대한 리스트를 유지한다.

- ITU-T Recommendation X.660 (2004), ‘Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: General procedure’.
- ITU-T Recommendation X.667, ‘Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: Generation and Registration of Universally Unique Identifiers (UUIDs) and their Use as ASN.1 Object Identifier Components’, 2004.
- ITU-T Recommendation X.680, ‘Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation’, 2002.
- ITU-T Recommendation X.800, ‘Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture’, 1991.
- ITU-T Recommendation X.811, ‘Security Frameworks for Open Systems: Authentication Framework’, 1995.
- ITU-T Recommendation X.812, ‘Security Frameworks for Open Systems: Access control framework’, 1995.
- ITU-T Recommendation X.1142, ‘Extensible Access Control Markup Language (XACML 2.0)’, 2006.
- IETF RFC 1034:1987, ‘Domain Names – Concepts and Facilities’, 1987.
- IETF RFC 1510:1993, ‘The Kerberos Network Authentication Requestor (V5)’, 1993.
- IETF RFC 1750:1994, ‘Randomness Recommendations for Security’, 1994.
- IETF RFC 1951:1996, ‘DEFLATE Compressed Data Format Specification Version 1.3’, 1996.
- IETF RFC 1991:1996, ‘PGP Message Exchange Formats’, 1996.
- IETF RFC 2045:1996, ‘Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message’, 1996.

- IETF RFC 2119:1997, 'Key words for use in RFCs to Indicate Requirement Levels', 1997.
- IETF RFC 2246:1999, 'The TLS Protocol Version 1.0', 1999.
- IETF RFC 2253:1997, 'Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished', 1997.
- IETF RFC 2396:1998, 'Uniform Resource Identifiers (URI): Generic Syntax', 1998.
- IETF RFC 2535:1999, 'Domain Name System Security Extensions', 1999.
- IETF RFC 2616 :1999, 'Hypertext Transfer Protocol – HTTP/1.1', 1999.
- IETF RFC 2617:1999, 'HTTP Authentication: Basic and Digest Access Authentication', 1999.
- IETF RFC 2798:2000, 'Definition of the inetOrgPerson LDAP Object Class', 2000.
- IETF RFC 2828:2000, 'Internet Security Glossary', 2000.
- IETF RFC 2914:2000 , 'Congestion Control Principles', 2000.
- IETF RFC 2915:2000, 'The Naming Authority Pointer (NAPTR) DNS Resource Record', 2000.
- IETF RFC 2945:2000, 'The SRP Authentication and Key Exchange System', 2000.
- IETF RFC 2965:2000, 'HTTP State Management Mechanism', 2000.
- IETF RFC 3061:2001, 'A URN Namespace of Object Identifiers', 2001.
- IETF RFC 3075:2001, 'XML-Signature Syntax and Processing', 2001.
- IETF RFC 3513:2003, 'Internet Protocol Version 6 (IPv6) Addressing Architecture', 2003.
- IETF RFC 3023:2001, 'XML Media Types', 2001.
- IETF RFC 3377:2002, 'Lightweight Directory Access Protocol (v3): Technical Specification', 2002.
- IETF RFC 3403:2002, 'Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database', 2002.
- IETF RFC 3546:2003, 'Transport Layer Security (TLS) Extensions', 2003.
- IETF RFC 3923:2004, 'End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP)', 2004.
- IETF RFC 4122:2005, 'A Universally Unique Identifier (UUID) URN Namespace', 2005.
- Liberty Alliance POAS:2003, R. 'Aarts, Reverse HTTP Binding for SOAP Specification Version 1.0, Liberty Alliance Project', 2003.
- OASIS WSS:2006, 'WS-Security Core Specification 1.1', February, 2006.

- UNICODE-C, M. Davis, M. J. Dürst, ,‘Dürst. Unicode Normalization Forms’. UNICODE Consortium, March 2001.
- W3C Canonicalization:2002, ,Exclusive XML Canonicalization Version 1.0’, W3C Recommendation, Copyright © World Wide Web Consortium, (Massachusetts Institute of Technology, Institute National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/xml-exc-c14n/>, 2002. July .2.
- W3C Character Model:2005,‘Character Model for the World Wide Web 1.0: Fundamentals’, W3C Recommendation, Copyright © [15 February 2005] World Wide Web Consortium, (Massachusetts Institute of Technology, Institute National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/2005/REC-charmod-20050215/>,2005.
- W3C Datatypes:2001, ‘XML Schema Part 2: Data types’, W3C Recommendation, Copyright © [2 May 2001] World Wide Web Consortium, (Massachusetts Institute of Technology, Institute National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>, 2001.
- W3C Encryption:2002, ‘XML Encryption Syntax and Processing’, W3C Recommendation, Copyright © [10 December 2002] World Wide Web Consortium, (Massachusetts Institute of Technology, Institute National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/>,2002.
- W3C Web Services Glossary:2004, ‘Web Services Glossary’, W3C Note, Copyright © [11 February 2004] World Wide Web Consortium, (Massachusetts Institute of Technology, Institute National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/ws-gloss/>,2004.
- W3C HTML:1999, HTML 4.01 Specification, ‘W3C Recommendation’, Copyright © [24 December 1999] World Wide Web Consortium, (Massachusetts Institute of Technology, Institute National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/REC-html40/>,1999.
- W3C Namespaces:1999, ,‘Namespaces in XML’, W3C Recommendation, Copyright © [14 January 1999] World Wide Web Consortium (Massachusetts Institute of Technology, Institute National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/REC-xml-names/>.,1999.

- W3C Primer:2005, 'SOAP Version 1.2 Part 0: Primer', W3C Recommendation, Copyright © [24 June 2005] World Wide Web Consortium, (Massachusetts Institute of Technology, Institute National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/2003/REC-soap12-part0-20030624/>,2005.
- W3C Signature:2002, 'XML Signature Syntax and Processing', W3C Recommendation, Copyright © [12 February 2002] World Wide Web Consortium, (Massachusetts Institute of Technology, Institute National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/xmlsigcore/>,2002.
- W3C Signature Schema:2001, 'XML Signature Schema, W3C Recommendation', Copyright © [1 March 2001] World Wide Web Consortium, (Massachusetts Institute of Technology, Institute National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/xmlsig-core/xmlsig-core-schema.xsd>, 2001.
- W3C String:1998, 'Requirements for String Identity Matching and String Indexing', W3C Note, Copyright © [10 July 1998] World Wide Web Consortium, (Massachusetts Institute of Technology, Institute National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/WD-charreq>,1998.
- W3C SOAP:2000, Simple Object Access Protocol (SOAP) 1.1, W3C Note, Copyright © [08 May 2000] World Wide Web Consortium, (Massachusetts Institute of Technology, Institute National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/2000/NOTE-SOAP-20000508>,2000.
- W3C XHTML:2002, The Extensible HyperText Markup Language (Second Edition), W3C Recommendation, Copyright © [1 August 2002] World Wide Web Consortium, (Massachusetts Institute of Technology, Institute National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/xhtml1/>,2002.
- W3C XML 1.0:2004, Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation, Copyright © [4 February 2004] World Wide Web Consortium, (Massachusetts Institute of Technology, Institute National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/REC-xml/>,2004.

- W3C XML Schema Part 1:2001, XML Schema Part 1: Structures, W3C Recommendation, Copyright © [2 May 2001] World Wide Web Consortium, (Massachusetts Institute of Technology, Institute National de Recherche en Informatique et en Automatique, Keio University), <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>,2001.

주의 - SAML 2.0 문서 내에 있는 문서에 대한 참조는 참조되는 문서의 상태를 제공하지는 않는다.

1.3. 용어 정의

SAML 2.0 에 대해, 다음과 같은 용어 정의가 적용된다.

1.3.1. 들여온 정의들(Imported definitions)

- SAML 2.0 은 ITU-T Rec. X.667 에서 정의된 다음 용어들을 사용한다:
 - a) UUID
- SAML 2.0 은 ITU-T Rec. X.680 에서 정의된 다음 용어들을 사용한다:
 - a) 객체 식별자(Object identifier);
 - b) 오픈 타입 표기법(Open type notation).
- SAML 2.0 은 ITU-T Rec. X.811 에서 정의된 다음 용어들을 사용한다:
 - 사용자(Principle).
- SAML 2.0 은 ITU-T Rec. X.812 에서 정의된 다음 용어들을 사용한다:
 - a) 접근 제어 정보(Access control information);
 - b) 사용자(User).
- SAML 2.0 은 W3C 웹 서비스 어휘에서 정의된 다음 용어들을 사용한다:
 - a) 초기 SOAP 송신자(Initial SOAP sender);
 - b) 네임스페이스(Namespace);
 - c) 최종 SOAP 수신자(Ultimate SOAP receiver);
 - d) XML 스키마(XML schema).

- SAML 2.0 은 IETF RFC 2828 에서 정의된 다음 용어들을 사용한다:
 - a) 접근(Access);
 - b) 접근 제어(Access control);
 - c) 프락시(Proxy);
 - d) 프락시 서버(Proxy server);
 - f) 풀(Pull);
 - e) 푸시(Push);
 - g) 보안 아키텍처(Security architecture);
 - h) 보안 정책(Security policy);
 - i) 보안 서비스(Security service).
- SAML 2.0 은 IETF RFC 2396 에서 정의된 다음 용어들을 사용한다:
 - a) Uniform resource identifier (URI);
 - b) URI 참조(URI reference).

1.3.2. 추가적인 용어정의(Additional definitions)

1.3.2.1. 접근 권한(Access rights)

주체가 자원에 대하여 가질수 있는 인가된 상호작용의 타입을 설명. 예로는 읽기, 쓰기, 실행, 추가, 변경 그리고 삭제를 들 수 있다.

1.3.2.2. 계정(Account)

사용자와 비즈니스 서비스 제공자 사이에 정상적인 거래와 서비스를 제공하기 위한 형식적인 비즈니스 협약.

1.3.2.3. 계정 연결(Account linkage)

서로 다른 두 제공자에서 동일한 사용자를 나타내는 계정을 연관시키는 방법. 이를 통해 두 제공자들은 그 사용자에 대한 정보를 통신할 수 있다. 계정 연결은 속성 공유나 또는 Identity 연계(federation)을 통해 설정될 수 있다.

1.3.2.3. 능동적인 역할(Active role)

예를 들어 자원에 접근하는 등, 어떤 연산을 수행할 때, 시스템 엔티티가 가지는 역할.

1.3.2.4. 관리 도메인(Administrative domain)

하나 또는 그 이상의 관리 정책, 인터넷 도메인 이름 등록들, 공공 법률 엔티티들(예를 들어, 개인, 기업 또는 다른 조직), 호스트, 네트워크 디바이스 그리고 상호 연결되는 네트워크의 집합, 그리고 그들 위에서 동작하는 네트워크 서비스와 응용들의 어떠한 조합으로 정의되는 환경 또는 문맥. 관리 도메인은 하나 또는 그 이상의 보안 도메인을 포함하거나 또는 정의할 수 있다. 하나의 관리 도메인은 단일한 사이트 또는 다중 사이트를 포함할 수 있다. 관리 도메인을 정의하는 특징들은 시간이 지남에 따라 진화할 수 있다. 관리 도메인들은 관리 도메인 경계를 넘어 서비스를 제공하거나 또는 소비하는 것에 대하여 협약을 만들 수 있다.

1.3.2.5. 관리자(Administrator)

시스템을 설치하거나 또는 관리하는 사람 또는 시스템을 이용하여 시스템 엔티티, 사용자와/또는 내용을 관리하는 사람. 관리자는 일반적으로 특정 관리 도메인에 가입하게 되고 하나 이상의 관리 도메인에 가입할 수도 있다.

1.3.2.6. 가맹, 가맹 그룹(Affiliation, affiliation Group)

사용자(principal)에 대한 식별자들의 (연계 관점에서) 단일한 네임스페이스를 공유하는 시스템 엔티티 집합.

1.3.2.7. 익명성(Anonymity)

익명 상태. 이것은 이름이나 신원이 알려지거나 노출되지 않도록 하는 조건을 나타냄.

1.3.2.8. 보장하는 기관(Asserting party)

공식적으로, 하나 또는 그 이상의 SAML 기관을 호스팅하는 관리 도메인. 비공식적으로, SAML 기관의 한 인스턴스.

1.3.2.9. 주장(Assertion)

주체에 대하여 수행되는 인증 행위, 주체에 대한 속성 정보 또는 명기된 자원에 대하여 주체가 행할 수 있는 인가 데이터 등에 대하여 SAML 기관이 생성한 데이터 조각.

1.3.2.10. 속성(Attribute)

객체의 독특한 특성. 실세계 객체에 대하여, 속성들은 종종 크기, 모양, 무게 및 색깔 등과 같은 물리적인 특징들로 명기된다. 사이버스페이스에서 객체는 크기, 인코딩 타입, 네트워크 주소 등등을 설명하는 속성들을 가질 수 있다. 속성들은 종종 “속성 이름”과 “속성 값(들)”으로 표현된다. 예를 들어, “foo”는 값 ‘bar’를 가지며, “count”는 값 1 을, “gizmo”는 ‘frob’과 ‘2’를 값들로 가진다.

1.3.2.11. 속성 주장(Attribute assertion)

주체의 속성들에 대한 정보를 운반하는 주장.

1.3.2.12. 속성기관(Attribute authority)

속성 주장들을 생성하는 시스템 엔티티.

1.3.2.13. 인증(Authentication)

인증은 어떤 사람 또는 어떤 사물이 어느 정도의 신뢰 내에서 그것이 자신이 그렇다고 선언하는 것이 정말로 맞는지 아닌지를 결정하는 과정이다.

1.3.2.14. 인증 주장(Authentication assertion)

주체에 대하여 발생된 성공적인 인증 행위에 대한 정보를 운반하는 주장.

1.3.2.15. 인증 기관(Authentication authority)

인증 주장들을 생성하는 시스템 엔티티.

1.3.2.16. 인가(Authorization)

어떤 주체가 특정 자원에 대하여 명기된 타입의 접근을 수행하는 것이 허가되었는지를, 적용가능한 접근제어 정보를 평가함으로써, 결정하는 과정. 일반적으로, 인가는 인증 문맥 내에 있다. 일단 주체가 인증이 되면, 그것은 다른 타입들의 접근을 수행하는 것에 대하여 인가될 수 있다.

1.3.2.17. 인가 결정(Authorization decision)

인가 행위의 결과. 그 결과는 부정적인 될 수 있다. 즉, 그것은 주체가 자원에 대하여 어떠한 접근 권한도 없음을 가리킨다.

1.3.2.18. 인가 결정 주장(Authorization decision assertion)

인가 결정에 대한 정보를 운반하는 주장.

1.3.2.19. 후 채널(Back channel)

후 채널은, 예를 들어 사용자 에이전트인 HTTP 클라이언트와 같은 또 다른 시스템 엔티티를 통하여 메시지를 리다이렉트(redirect) 하지 않고 두 시스템 엔티티들 사이에 직접적인 통신을 가리킨다.

1.3.2.20. 바인딩, 프로토콜 바인딩(Binding, protocol binding)

일반적으로, 어떤 프로토콜 메시지와 메시지 교환 패턴을 구체적인 방식으로 또 다른 프로토콜로 매핑시키는 것에 대한 명세임. 예를 들어, SAML <AuthnRequest> 메시지를 HTTP 에 매핑하는 것은 바인딩의 한 예가 된다. 동일한 SAML 메시지를 SOAP 으로 매핑하는 것은 또 다른 바인딩이 된다. SAML 문맥에서는, 각각의 바인딩에 “SAML xxx binding”이라는 패턴의 이름이 주어진다.

1.3.2.21. 크리덴셜(Credentials)

주장되는 사용자(principal) 신원을 확인하기 위해 전송되는 데이터.

1.3.2.22. 최종 사용자(End user)

응용 목적으로 자원을 사용하는 자연인(natural person).

1.3.2.23. 엔티티(Entity)

“시스템 엔티티”를 참고한다.

1.3.2.24. 연계하다(Federate)

둘 또는 그 이상의 엔티티들을 함께 연결하거나 또는 바인딩하기.

1.3.2.25. 연계(Federation)

이 용어는 두가지 의미로 사용된다.:

1. 두 엔티티 사이에 관계를 설정하는 행위.
2. 어떠한 개수의 서비스 제공자들과 아이덴티티 제공자들로 구성된 하나의 연합(association).

1.3.2.26. 연계된 아이덴티티(Federated identity)

제공자들 사이에 그 사용자를 참조하기 위해 사용되는 식별자 집합과 속성들에 대하여 협정(agreement)가 있을 때, 사용자(principal)의 아이덴티티는 연계가 되었다고 말해진다.

1.3.2.27. 전 채널(Front channel)

전 채널은 두 개의 HTTP 로 통신하는 서버들이 “HTTP redirect” 메시지를 채용하고 이를 통해, 예를 들어 웹 브라우저 또는 다른 어떠한 HTTP 클라이언트인 사용자 에이전트를 경유하여 상호간에 메시지를 전달하는 경우에 효과가 발생하는 통신 채널을 가리킨다.

1.3.2.28. 식별자(Identifier)

시스템 엔티티들 유일하게 가리키도록 시스템 엔티티에 매핑된 데이터 객체. 예를 들어 문자열이 될 수 있음. 시스템 엔티티는 그것을 가리키는 다중 식별자를 가질 수 있다. 하나의 식별자는 본질적으로 엔티티의 “구별되는 속성”이다.

1.3.2.29. 아이덴티티, 신원(Identity)

엔티티의 본질. 어떤 사물의 아이덴티티는 어떤 사물의 특징들로 종종 설명된다. 이 특성들 중에 식별자들이 포함될 수 있다.

1.3.2.30. 아이덴티티 탈연계(Identity defederation)

제공자들이 일정 집합의 식별자와/또는 속성들을 통해 사용자(principal)을 참조하는 것을 그만두기로 동의할 때, 발생하는 동작.

1.3.2.31. 아이덴티티 연계(Identity federation)

사용자(principal)을 위해 연계된 아이덴티티를 생성하는 동작.

1.3.2.32. 아이덴티티 제공자(Identity provider)

사용자(principal)들을 위해 아이덴티티 정보를 생성하고, 유지하며, 관리하고 그리고 웹 브라우저 프로파일과 같이 하나의 연계 내에서 다른 서비스 제공자에게 사용자(principal) 인증을 제공하는 일종의 서비스 제공자.

1.3.2.33. 아이덴티티 제공자 라이트(Identity provider lite)

단지 SAML 에서 요구되는 부분만을 사용하여, 사용자(principal)들을 위해 아이덴티티 정보를 생성하고, 유지하며, 관리하고 그리고 웹 브라우저 프로파일과 같이 하나의 연계 내에서 다른 서비스 제공자에게 사용자(principal) 인증을 제공하는 일종의 서비스 제공자.

1.3.2.34. 로그인, 로그온, 사인-온(Login, logon, sign-on)

일종의 처리. 이 처리를 통해 사용자가 인증기관에게 크리덴셜을 제출하고 간단한 세션을 설정하고 그리고 선택적으로 리치(rich) 세션을 설정한다.

1.3.2.35. 로그아웃, 로그오프, 사인-오프(Logout, logoff, sign-off)

일종의 처리. 이 처리를 통해 사용자는 단순 세션 또는 리치(rich) 세션을 종료하기를 원한다는 것을 알린다.

1.3.2.36. 마크업 언어(Markup language)

특수한 목적으로 XML 문서의 구조에 적용되는 일단의 XML 요소들과 XML 속성들. 마크업 언어는 일반적으로 일단의 XML 스키마들과 동반되는 문서로 정의된다.

1.3.2.37. 이름 제한자(Name qualifier)

다른 사용자들(principals)을 나타내기 위해, (연계 관점에서) 하나 이상의 네임스페이스에서 사용될 수 있는 하나의 식별자가 모호해지지 않도록 해 주는 문자열.

1.3.2.38. 기관, 당사자(Party)

비공식적으로, 주장을 수신하거나 또는 자원을 접근하는 것과 같은 어떤 처리나 통신에 참여하는 하나 또는 그 이상의 사용자들(principals).

1.3.2.39. 영속적인 의사익명(Persistent pseudonym)

다중 세션에 걸쳐있는 확장된 기간 동안에 주어진 의지하는 기관이 사용자를 식별할 수 있도록, 어떤 아이덴티티 제공자에 의해 할당된 프라이버시-보호형 이름 식별자. 아이덴티티 연계를 나타내는데 사용될 수 있다.

1.3.2.40. 정책 결정점(Policy decision point (PDP))

자신을 위해 인가 결정을 내리거나 또는 이와 같은 결정을 요구하는 다른 시스템 엔티티를 위해 인가 결정을 내리는 시스템 엔티티. 예를 들어, SAML PDP 는 인가 결정 요청들을 받아들여, 응답으로 인가 결정 주장들을 생성한다. PDP 는 인가 결정 기관이다.

1.3.2.41. 정책 집행점(Policy enforcement point (PEP))

인가 결정을 요청하고 뒤이어 집행하는 시스템 엔티티. 예를 들어, SAML PEP 는 일가 결정 요청들을 PDP 에게 전달하고, 응답으로 수신되는 인가 결정 주장들을 처리한다.

1.3.2.42. 사용자 아이덴티티(Principal identity)

일반적으로 식별자인 어떤 사용자 아이덴티티의 표현.

1.3.2.43. 프로파일(Profile)

여러 목적 중에 하나를 위한 일단의 규칙들. 각각의 집합은 “SAML xxx 프로파일” 또는 “xxx SAML 프로파일” 패턴으로 이름이 주어진다.

- 어떤 프로토콜 또는 다른 사용 문맥에 주장을 내장시키거나 또는 그것들로부터 추출하는 방법에 대한 규칙들.
- 특수한 사용 문맥에서 SAML 프로토콜 메시지를 사용하는 것에 대한 규칙들.
- SAML 로 표현된 속성들을 또 다른 속성 표현 시스템으로 매핑시키는 것에 대한 규칙들. 이와 같은 규칙의 집합은 “속성 프로파일”로 알려진다.

1.3.2.44. 프로토콜 바인딩(Protocol binding)

“바인딩”을 참고한다.

1.3.2.45. 제공자(Provider)

아이덴티티 제공자들과 서비스 제공자들 둘 다를 가리키는 포괄적인 표현.

1.3.2.46. 의지하는 기관(측)(Relying party)

다른 시스템 엔티티가 제공한 정보를 기반으로 행동을 취할 것을 결정하는 시스템 엔티티. 예를 들어, SAML 의지하는 기관은 주체에 대하여 보장하는 기관(SAML 기관)이 제공한 주장들을 의지한다.

1.3.2.47. 요청자(Requester)

또 다른 시스템 엔티티(SAML 기관, 응답자)에게 서비스를 요청하기 위해 SAML 프로토콜을 활용하는 시스템 엔티티. 많은 시스템 엔티티들이 클라이언트와 서버 둘 모두로서 동시에 또는 순차적으로 동작하기 때문에, 이 표시법에서 “클라이언트” 라는 용어는 사용되지 안 된다. SAML SOAP 바인딩이 사용중인 경우에는, SAML 요청자는 초기 SOAP 송신자와 구조적으로 분리된다.

1.3.2.48. 자원(Resource)

(예를 들어, 파일 형태나 메모리 형태, 등등으로) 하나의 정보 시스템에 포함되는 데이터, 또한:

- 시스템이 제공하는 서비스.
- 시스템 장비의 한 항목(다른 말로, 하드웨어, 펌웨어, 소프트웨어 또는 문서등과 같은 시스템 컴포넌트)

1.3.2.49. 응답자(Responder)

또 다른 시스템 엔티티(요청자)로부터 전달받은 서비스 요청에 대하여 응답하기 위해 SAML 프로토콜을 활용하는 시스템 엔티티(SAML 기관). 많은 시스템 엔티티들이 클라이언트와 서버 둘 모두로서 동시에 또는 순차적으로 동작하기 때문에, 이 표시법에서 “서버” 라는 용어는 사용되지 안 된다. SAML SOAP 바인딩이 사용중인 경우에는, SAML 응답자는 최종 SOAP 수신자와 구조적으로 분리된다.

1.3.2.50. 역할, 룰(Role)

사전들은 역할을 “수행자에 의해 동작되는 특성” 또는 “함수 또는 위치)로 정의한다. 시스템 엔티티들은 예를 들어 능동적인 역할들과 수동적인 역할들과 같은 다양한 타입들의 역할들을 순차적으로/또는 동시적으로 수행한다. 관리자의 개념은 종종 역할의 한 예이다.

1.3.2.51. SAML 아티팩트(SAML artifact)

일반적으로 더 크고, 가변-크기의 SAML 프로토콜 메시지를 가리키는 작고, 고정-크기를 가지는 구조화된 데이터 객체. SAML 아티팩트들은 “3xx Redirection” 상태 코드들을 가지는 HTTP 응답 메시지들과 뒤따르는 HTTP GET 메시지들과 같이 URL 에 내장되고 HTTP 메시지를 통해 운반되도록 설계된다. 이런 방식으로, 서비스 제공자는 간접적으로, 사용자 에이전트를 경유하여, 다른 제공자에게 SAML 아티팩트를 전달할 수 있다. 다른 제공자는 artifact 를 제공하는 제공자와의 직접적인 상호작용을 통해 SAML 아티팩트를 디레퍼런스(dereference)하여 SAML 프로토콜 메시지를 얻을 수 있다.

1.3.2.52. SAML 기관(SAML authority)

SAML 도메인 모델에서 주장들을 발급하는 추상적인 시스템 엔티티. 속성 기관, 인증 기관, 정책 결정점(PDP)를 또한 참고한다.

1.3.2.53. 보안(Security)

정보의 기밀성을 보장하고, 그것을 처리하는데 사용되는 시스템과 네트워크를 보호하고, 그들에 대한 접근을 제어하는 일단의 보호방법들. 보안은 일반적으로 비밀(secretcy), 기밀성, 무결성, 이용가능성 등의 개념을 포괄한다. 이것은 어떤 시스템이 잠재적으로 상호연관된 공격들을 방어하는 것을 보장하기 위한 것이다.

1.3.2.54. 보안 주장(Security assertion)

보안 아키텍처의 문맥에서 철저히 검사된 주장.

1.3.2.55. 보안 문맥(Security context)

개별적인 SAML 프로토콜 메시지에 대하여, 메시지의 보안 문맥은 만약 있다면 메시지의 보안 헤더 블록들과 수신자에게 메시지를 배달할 때, 사용될 수 있는 다른 보안 메커니즘들의 의미적인 합(semantic union)이다. HTTP, TLS 와 IPSEC 등과 같은 하부 네트워크 스택 레이어들에서 채택되는 보안 메커니즘들이 후자의 예가 된다.

1.3.2.56. 보안 도메인(Security domain)

일단의 자원들과 그들 자원들을 접근하는 것이 인가된 시스템 엔티티들을 포함하여, 보안 모델과 보안 아키텍처에서 정의된 환경 또는 문맥. 하나 또는 그 이상의 보안 도메인들이 단일 관리 도메인(administrative domain)에 존재할 수 있다. 어떠한 보안 도메인을 정의하는 특징들은 시간이 지남에 따라 일반적으로 진화한다.

1.3.2.57. 보안 정책 표현(Security policy expression)

사용자(principal) 아이덴티티들과 또는 그것의 속성들을 허용가능한 동작들(actions)로 매핑하는 것. 보안 정책 표현은 종종 본질적으로 접근 제어 리스트가 된다.

1.3.2.58. 서비스 제공자(Service provider)

어떤 시스템 엔티티에게 주어진 역할. 이 역할을 통해 그 시스템 엔티티는 사용자들(principals) 또는 다른 시스템 엔티티들에게 서비스들을 제공한다.

1.3.2.59. 서비스 제공자 라이트(Service provider lite)

어떤 시스템 엔티티에게 주어진 역할. 이 역할을 통해 그 시스템 엔티티는 단지 필요한 SAML 프로토콜 부분만을 사용하여, 사용자들(principals) 또는 다른 시스템 엔티티들에게 서비스들을 제공한다.

1.3.2.60. 세션(Session)

상호작용 기간 동안 상호작용에 대한 일부 상태를 유지하는 것을 특징으로 하는, 종종 사용자를 포함하는(Principal), 시스템 엔티티들의 지속적인 상호작용.

1.3.2.61. 세션 기관(Session authority)

세션들과 관련된 상태를 어떤 시스템 엔티티가 유지할 때, 그 기관에게 주어진 역할.

1.3.2.62. 세션 참여자(Session participant)

어떤 기관이 적어도 하나의 세션 기관과 어떤 세션에 참여할 때, 그 기관에게 주어진 역할.

1.3.2.63. 사인-오프(Sign-off)

“로그아웃”을 참고한다.

1.3.2.64. 사인-온(Sign-on)

“로그인”을 참고한다.

1.3.2.65. 사이트(Site)

지리적인 또는 DNS 이름 관점에서 하나의 관리 도메인을 나타내는 비공식적인 용어. 이것은 어떤 관리 도메인의 특정 지리적인 또는 위상적인(topological) 부분을 나타낼 수도 있고, 또는 하나의 ASP 사이트에서 그렇듯이, 다중 관리 도메인들을 포괄할 수도 있다.

1.3.2.66. 주체(Subject)

어떤 보안 도메인 문맥에서 하나의 사용자(principal). SAML 주장들은 주체에 대한 선언들을 생성한다.

1.3.2.67. 시스템 엔티티, 엔티티(System entity, entity)

컴퓨터/네트워크 시스템의 능동적인 어떤 요소. 예를 들어, 자동화된 처리 또는 처리 집합. 하부 시스템, 분리된 기능 집합을 통합하는 사람 또는 사람들 그룹.

1.3.2.68. 타임-아웃(Time-out)

만약 어떤 사건이 발생하지 않았다면, 그 시각 이후, 어떤 조건이 “참”이 되는 기간. 예를 들어, 세션의 상태가 특정 기간 동안 비활성화되어 있었기 때문에 종료되는 세션은 “타임 아웃” 되었다고 말해진다.

1.3.2.69. 일시적인 의사익명(Transient pseudonym)

다중 세션에 걸쳐있을 필요가 없는 상대적으로 짧은 기간 동안에 주어진 의지하는 기관이 사용자를 식별할 수 있도록, 어떤 아이덴티티 제공자에 의해 할당된 프라이버시-보호형 이름 식별자.

1.3.2.70. XML 요소(XML attribute)

XML 요소의 시작-태그(start-tag)에 포함되어 있고, 이름과 값을 가지는 XML 데이터 구조.

1.3.2.71. XML 요소(XML element)

XML 문서 내에서 다른 이와 같은 구조들 사이에서 구조적으로 배열되며, 시작-태그(start-tag)와 종료-태그(end-tag) 또는 빈 태그(empty tag)로 가리켜지는 XML 데이터 구조.

1.4. 약어(Abbreviations)

AA	Attribute Authority
ASN.1	Abstract Syntax Notation One
ASP	Application Service Provider
CA	Certification Authority
CMP	Certificate Management Protocol
CRL	Certificate Revocation List
DDDS	Dynamic Delegation Discovery System
DCE	Distributed Computing Environment
DNS	Domain Name System

ECP	Enhanced Client/Proxy
HTTP	HyperText Transfer Protocol
HTTPS	Secure HyperText Transport Protocol
IdP	Identity Provider
IdP Lite	Identity Provider Lite
IP	Internet Protocol
IPSEC	Internet Protocol SECurity
MD5	Message Digest algorithm 5
MIME	Multipurpose Internet Mail Extensions
NAPTR	Naming Authority PoinTeR
OID	Object IDentifier
PAC	Privilege Attribute Certificates
PAOS	Reverse SOAP
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PGP	Pretty Good Privacy
PKI	Public-Key Infrastructure
POP	Proof Of Possession
RA	Registration Authority
RSA	Rivest Shamir Adleman public key algorithm
SHA-1	Secure Hash Algorithm 1
SP	Service Provider
SPKI	Simple Public Key Infrastructure
SP Lite	Service Provider Lite
SSO	Single Sign On
TLS	Transport Layer Security protocol
URI	Uniform Resource Identifier
UTC	Coordinated Universal Time
UUID	Universal Unique IDentifier
XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language

1.5. 관례(Conventions)

SAML 2.0 에서 사용되는 키워드인 "해야만 한다(must)", "하지 않아야만 한다(must not)", "요구된다(required)", "일 것이다(shall)", "이지 않을 것이다(shall not)", "해야 한다(should)", "하지 않아야 한다(should not)", "권고된다(recommended)", "일(할) 수 있다(may)", "선택적인(optional)" 은 IETF RFC 2119 에서 설명된 것과 같이 해석되어야 한다.

SAML 2.0 은 W3C XML 스키마 Part 1, W3C 스키마 Part 2 와 그들 표준들의 규범적 텍스트(normative text)를 사용하여 XML 인코딩된 SAML 주장과 프로토콜 메시지들의 문법과 의미를 설명한다. SAML 2.0 의 SAML 스키마 문서들과 스키마 리스트 사이에 불일치가 발생할 경우에는, 스키마 문서가 높은 우선순위를 가진다. 어떤 경우에는, SAML 2.0 이 스키마 문서에 의해 가리키는 것 이상의 제약을 가하는 경우가 있다는 것에 주의해야 한다.

1.6. 공통 데이터 타입(Common data types)

다음 하부 절들은 SAML 스키마들에서 나타나는 공통된 데이터 타입들을 어떻게 사용하고 해석하는지를 정의한다.

1.6.1. 문자열 값(String Values)

모든 SAML 문자열 값들은 **xs:string** 타입을 가지며, 이 타입은 W3C XML 스키마 데이터타입들 표준에 내장(built in) 되어 있다. SAML 2.0 에서 별다른 언급이 없으면, SAML 메시지들에 존재하는 모든 문자열들은 적어도 하나 이상의 공백이 아닌 문자(non-whitespace)로 구성되어야만 한다.

이 SAML 2.0 또는 특정 프로파일들에서 별다른 언급이 없으면, XML 스키마 **xs:string** 타입을 가지거나 또는 이 문자열 타입으로부터 유도된 타입을 가지는 SAML 문서 내의 모든 요소들은 정확한 이진 비교(exact binary comparison)를 사용하여 비교되어야만 한다. 특히, SAML 구현과 배치(deployment)들은 대소문자를 구분하지 않는 문자열 비교, 공백의 정규화 또는 절단(trimming) 또는 숫자나 화폐와 같이 로케일에 따라 고유한(locale-specific) 변환 등에 의존하지 않아야만 한다. 이 요구는 W3C 문자열의 요구사항을 따르게 하기 위해 의도된 것이다.

만약 어떤 구현이 다른 문자 인코딩(encodings) 방식들을 사용하여 표현된 값들을 비교한다면, 그 구현은 두 값을 유니코드 문자 인코딩인 정규화 폼 C(Normalization Form C)로 변환하고 그것들에 대하여 정확한 이진 비교를 수행한 것과 같은 결과를 반환하는 비교 방법을 사용해야만 한다. 이 요구는 W3C 문자 모델과 특히, 유니코드-정규화 텍스트(Unicode-normalized Text)들에 대한 규칙을 따르게 하기 위해 의도된 것이다.

SAML 문서 형태로 받은 데이터와 외부 소스로부터 받은 데이터를 비교하는 응용(application)은 XML 에 대해 규정된 정규화 규칙을 고려해야만 한다. 요소들 내에 포함된 텍스트(text)는 라인의 끝이 라인피드 문자들(ASCII code 10_{Decimal})을 사용하여 나타내도록 정규화된다. 문자열들 (또는 문자열로부터 유도된 타입들)로 정의된 XML 속성 값들은 W3C XML 1.0, 3.3.3 절에서 설명된 것처럼 정규화된다. 모든 공백 문자들은 스페이스(blanks) (ASCII code 32_{Decimal})로 대체된다.

SAML 2.0 은 XML 속성 값들 또는 요소 내용에 대하여 대조(collation) 또는 정렬 순서를 정의하지 않는다. SAML 구현들은 값들에 대하여 특정한 정렬 순서들에 의존하지 않아야만 한다. 왜냐하면 처리에 참여한 호스트(host)들에서 설정된 로케일(locale)에 따라, 그 정렬 순서들이 달라지기 때문이다.

1.6.2. URI 값(URI Values)

모든 SAML URI 참조 값들은 **xs:anyURI** 타입을 가지며, 이 타입은 W3C XML 스키마 데이터타입들에 내장(built in) 되어 있다.

SAML 2.0 에서 다르게 지시되지 않는다면, SAML 에서 정의된 속성들 또는 요소들 내에서 사용되는 모든 URI 참조 값들은 적어도 하나 이상의 공백이 아닌 문자로 구성되어야만 하며, 절대경로를 표현하도록 요구된다.

SAML 2.0 은 상태코드, 포맷 타입, 속성과 시스템 엔티티 이름들 등과 같은 식별자들로써 URI 참조를 광범위하게 사용한다. 따라서, 똑 같은 URI 가 다른 시각에 다른 정보를 나타내는데 절대로 사용되지 않도록, URI 값들이 유일하고 동시에 일관되도록(consistent) 하는 것이 필수적이다.

1.6.3. 시간 값(Time Value)

모든 SAML 의 시각 값들은 **xs:dateTime** 타입을 가지며, 이 타입은 W3C XML 스키마 데이터타입들에 내장(built in) 되어 있다. 모든 SAML 시각 값들은 시간대(time zone) 컴포넌트가 없는 UTC 형식(form)으로 표현되어야만 한다.

SAML 시스템 엔티티들은 1000 분의 1 초보다 더 정교한 시각에 의존하지 않아야 한다. 구현들은 윤초(leap seconds)를 명기하는 시각 값들을 생성하지 않아야만 한다.

1.6.4. ID 와 ID 참조 값(ID and ID Reference Values)

xs:ID 단순 타입은 주장들, 요청 및 응답에 대한 SAML 식별자들(identifiers)을 선언하는데 사용된다. SAML 2.0 에서 **xs:ID** 타입으로 선언된 값들은 **xs:ID** 타입 자체의 정의에 의해 주어진 특성뿐만 아니라 다음과 같은 특성들을 만족시켜야만 한다:

- 식별자들은 할당하는 어떠한 기관(party)도 자신 또는 다른 기관(party)이 다른 데이터 객체에게 우연히 동일한 식별자를 할당할 수 있는 가능성이 거의 무시할 수 있을 정도라는 것을 보장해야만 한다.
- 어떤 데이터 객체가 자신이 특정한 식별자를 가지고 있다고 선언한 곳에, 그와 같은 선언은 정확히 하나만 있어야만 한다.

SAML 시스템 엔티티가 그것이 생성하는 식별자가 유일하다는 것을 보장하는 메커니즘은 시스템 구현에 의해 결정된다. 랜덤(random) 또는 의사랜덤(pseudorandom) 기술이 채택된 경우에, 임의적으로 선택된 두 개의 식별자가 서로 동일할 확률은 2^{-128} 보다 작거나 같아야만 하고, 2^{-160} 보다 작거나 같아야 한다. 이 요구는 128 비트와 160 비트 사이의 길이를 갖는 임의적으로 선택된 값을 인코딩함으로써 충족될 수 있다. 인코딩은 **xs:ID** 데이터타입을 정의하는 규칙을 준용해야만 한다. 의사랜덤 발생기는 서로 다른 시스템들 사이에 바람직한 유일성 특성을 보장하기 위해 유일한 값(material)으로 시드(seed)를 설정하여야만 한다.

xs:NCName 단순 타입은 SAML 에서 **xs:ID** 타입의 식별자들을 참조하는데 사용된다. 이렇게 하는 이유는 **xs:IDREF** 가 이런 목적으로 사용될 수 없기 때문이다. SAML 에서, SAML 식별자 참조에 의해 참조되는 요소는 식별자 참조가 사용되는 문서와 다른 문서에서 실질적으로 정의될 수 있다. **xs:IDREF** 를 사용하게 되면, 그것의 값이 동일한 XML 문서에 있는 어떤 요소의 ID 속성 값과 매치(match) 되어야 한다는 요구를 위반하게 될 것이다.

2. SAML 주장

SAML 은 시스템 엔티티가 어떤 주체에 대하여 생성한 주장의 문법과 처리 규칙을 정의한다. 이와 같은 주장을 만들거나 또는 의지하기 위해, SAML 시스템 엔티티들은 주장 자체 또는 주장의 주체에 대한 내용을 통신하기 위해 다른 프로토콜을 사용할 수 있다. 이 표준은 SAML 보장의 구조, 관련된 프로토콜 집합, 그리고 SAML 시스템을 관리하는데 관련된 처리 규칙들을 정의한다.

SAML 주장과 프로토콜 메시지들은 XML(W3C XML 1.0 을 참조)로 인코딩되어 있으며, XML 네임스페이스(W3C Namespaces 참조)를 사용한다. 이것들은 일반적으로 HTTP POST 또는 XML 로 인코딩된 SOAP 메시지와 같은 전송을 위한 다른 구조에 내장된다. SAML 바인딩은 SAML 프로토콜 메시지들을 내장하고 전송하기 위한 프레임워크를 제공한다. SAML 프로파일은 SAML 특징들을 사용할 때, 특정 사용예(use case)를 달성하고 상호운용성을 달성하기 위해, SAML 주장과 프로토콜을 어떻게 사용해야 하는지에 대한 기본 프로파일 집합을 제공한다.

주장은 SAML 기관(SAML authority)에 의해 만들어지는 영 또는 그 이상의 문장(statement)들을 제공하는 정보의 패키지(package)이다; SAML 기관들은 때때로 주장 생성과 교환을 논의(discussion)할 때 보장하는 기관(asserting party)으로 언급된다. 수신된 주장들을 사용하는 시스템 엔티티들은 의지하는 기관(relying party)으로 알려진다. (이러한 용어들은 SAML 프로토콜 메시지 교환의 토론을 위해 예약되어(reserved) 있는 요청자(requester)와 응답자(responder) 용어와는 다르다는 것에 주의해야 한다.)

SAML 주장들은 일반적으로 <Subject> 요소로 나타나는 주체(subject)에 대하여 만들어진다. 그러나 <Subject> 요소는 선택요소이고, 다른 표준들이나 프로파일들은 주체를 명기하지 않거나 다른 방식으로 주체를 명기하면서 유사한 문장을 만들기 위해 SAML 주장 구조를 활용할 수도 있다. 일반적으로 많은 서비스 제공자들은 주체에 대한 주장을 사용하여 접근을 제어하고 맞춤형(customized) 서비스를 제공한다. 따라서 서비스 제공자들은 IdP(identity provider, 신원 제공자)로 불리는 보장하는 기관을 의지하는 기관이 된다.

이 표준은 SAML 기관에서 생성될 수 있는 세가지 종류의 문장(statement)을 정의한다. 모든 SAML 에서 정의된 문장들은 주체와 연관된다. 이 표준에서 정의된 세가지 종류의 문장들은 다음과 같다.

- 인증(Authentication): 주장 주체가 특정한 시각에 특정한 방식으로 인증되었다.
- 속성(Attribute): 주장 주체가 제공되는 속성들과 연관되어 있다.
- 인가 결정(Authorization Decision): 주장 주체가 특정 자원을 접근하려는 요구가 허가되었거나 또는 거절되었다.

주의: PE13(OASIS PE:2006 참조)는 위 문단에 “또는 미결정(indeterminate)”를 추가할 것을 제안한다.

주장의 바깥 구조는 일반적이며 주장 내의 모든 문장들에 공통적인 정보를 제공한다. 주장 내에서, 일련의 내부 요소들은 인증, 속성, 인가 결정 또는 특수한 내용을 포함하는 사용자 정의 문장들을 설명한다.

SAML 주장 스키마에서는 확장(extensions)이 허용되며, 이것은 주장과 문장에 사용자가 정의한 확장들이 허용되게 하며 새로운 종류의 주장들과 문장들의 정의가 허용된다.

2.1. 스키마 헤더와 네임스페이스 선언 (Schema Header and Namespace Declarations)

다음의 스키마 조각(fragment)은 주장 스키마를 위한 XML 네임스페이스들과 다른 헤더 정보들을 정의한다.

```

<schema targetNamespace="urn:oasis:names:tc:SAML:2.0:assertion"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  elementFormDefault="unqualified"
  attributeFormDefault="unqualified"
  blockDefault="substitution"
  version="2.0">
  <import namespace="http://www.w3.org/2000/09/xmldsig#"
    schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
      20020212/xmldsig-core-schema.xsd"/>
  <import namespace="http://www.w3.org/2001/04/xmlenc#"
    schemaLocation="http://www.w3.org/TR/2002/REC-xmlenc-core-
      20021210/xenc-schema.xsd"/>
  <annotation>
    <documentation>
      Document identifier: saml-schema-assertion-2.0
      Location: http://docs.oasis-open.org/security/saml/v2.0/
      Revision history:
        V1.0 (November, 2002):
          Initial Standard Schema.
        V1.1 (September, 2003):
          Updates within the same V1.0 namespace.
        V2.0 (March, 2005):
          New assertion schema for SAML V2.0 namespace.
    </documentation>
  </annotation>
  ...
</schema>

```

2.2. 이름 식별자(Name Identifier)

이번 절은 주체에 대한 주장들과 프로토콜 메시지들의 주체들과 발급자들(issuers)에 대하여 설명적인 식별자들을 포함하는 SAML 구조들을 정의한다.

SAML 에서는 두 개의 시스템 엔티티들이 제 3 자(third party)에 관하여 통신하는 것이 유용한 경우가 많이 있다. 예를 들어, SAML 인증 요청 프로토콜은 제 3 자가 주체를 인증할 수 있도록 한다. 따라서, 이것은 기관들이 기관들 각각에 의미 있는 식별자들로 연관시킬 수 있는 수단을 확립하는데 유용하다. 일부 경우에는, 하나의 식별자가 작은 집단의 시스템 엔티티에서만 사용되도록 영역을 제한하는 것이 필요할 것이다(예를 들어, 주체의 프라이버시를 보호하기 위해). 유사한 식별자들은 또한 SAML 프로토콜 메시지 또는 주장들의 발급자를 참조하는데 사용될 수 있다.

두 개 또는 그 이상의 시스템 엔티티들이 다른 식별들(identities)을 참조할 때, 동일한 이름 식별자 값을 사용할 가능성이 있다. 따라서, 각각의 엔티티는 동일한 이름에 대해 각기 다르게 이해할 수도 있다. SAML 은 이름 제한자(name qualifier)들과 관련된 연계된 네임스페이스(federated namespace)에 이름 식별자를 효과적으로 위치시킴으로써 이름 식별자가 혼동되는 것을 방지하는 이름 제한자(name qualifier)를 제공한다. SAML 2.0 은 보장하는 기관과 특정한 의지하는 기관 또는 연합기관(affiliation)들 모두에서 식별자가 제한되는 것을 허용하고, 이렇게 함으로써, 필요할 때, 식별자들이 쌍방향 의미를 표시할 수 있도록 한다.

이름 식별자는 또한, 프라이버시 보호 측면을 강화하기 위해 암호화될 수 있으며, 특히, 중개자(intermediary)를 통해 식별자들이 전달되는 환경에서는 더욱 더 암호화될 수 있다.

주의: (여러 이유들 중에) 상대적으로 개선된 XML 스키마 구조들의 사용을 피하기 위해, 다양한 타입의 식별자 요소들은 공통 타입 계층(type hierarchy)를 공유하지 않는다.

2.2.1. <BaseID> 요소

<BaseID> 요소는 어플리케이션들이 새로운 종류의 식별자들을 추가할 수 있도록 허용하는 확장점(extension point)이다. 이 요소의 BaseIDAbstractType 복합 타입은 추상타입이고 따라서 단지 유도 타입의 기반(base)으로만 사용될 수 있다. 이 요소는 단지 확장된 식별자 표현을 위해 사용되며 다음 속성들을 포함한다:

NameQualifier [Optional]

식별자를 제한하는 보안 또는 관리 도메인. 이 속성은 이질적인(disparate) 사용자 저장소(stores)로부터 식별자들은 연계(federate)하는 수단을 제공한다.

SPNameQualifier [Optional]

서비스 제공자 또는 제공자들의 연합기관(affiliation)의 이름으로 식별자를 더욱 한정한다. 이 속성은 의지하는 기관(들) (relying party or relying parties)을 기초로 하여 식별자를 연계하는 추가적인 수단을 제공한다.

만약 식별자의 타입 정의가 명시적으로 그들의 사용과 의미를 정의하고 있지 않다면, NameQualifier 와 SPNameQualifier 속성들은 생략해야 한다.

다음 스키마 조각은 <BaseID> 요소와 **BaseIDAbstractType** 복합 타입을 정의한다.

```
<attributeGroup name="IDNameQualifiers">
  <attribute name="NameQualifier" type="string" use="optional"/>
  <attribute name="SPNameQualifier" type="string" use="optional"/>
</attributeGroup>

<element name="BaseID" type="saml:BaseIDAbstractType"/>
<complexType name="BaseIDAbstractType" abstract="true">
  <attributeGroup ref="saml:IDNameQualifiers"/>
</complexType>
```

2.2.2. NameIDType 복합 타입

NameIDType 복합 타입은 요소가 문자열 값을 가지는 이름으로 엔티티를 나타낼 때 사용된다. 이 타입은 <BaseID> 요소보다 더 제약적인 형태의 식별자이며, <NameID>와 <Issuer> 요소들의 기반이 되는 타입이다. 실질적인 식별자를 포함하는 문자열 내용뿐만 아니라, 이 타입은 다음 선택적인 속성들을 제공한다:

NameQualifier [Optional]

식별자를 제한하는 보안 또는 관리 도메인. 이 속성은 이질적인(disparate) 사용자 저장소(stores)로부터 식별자들은 연계(federate)하는 수단을 제공한다.

SPNameQualifier [Optional]

서비스 제공자 또는 제공자들의 연합기관(affiliation)의 이름으로 식별자를 더욱 더 한정한다. 이 속성은 의지하는 기관(들) (relying party or relying parties)을 기초로 하여 식별자를 연계하는 추가적인 수단을 제공한다.

Format [Optional]

문자열 기반의 식별자에 대한 분류(classification)를 나타내는 URI 참조. Format 속성의 값으로 사용될 수 있는 SAML 에서 정의된 URI 참조들과 관련된 설명 및 처리 규칙은 0 절을 참조한다. 이 타입에 기반한 요소에 의해 다르게 명기되지 않았다면, 그리고 만약 어떠한 Format 값도 제공되지 않는다면, urn:oasis:names:tc:SAML:1.0:nameid-format:unspecified (8.3.1 절을 참조) 값이 효과를 갖게 된다.

8.3 절에서 명기된 것과 다른 Format 값이 사용될 때, 이 타입 요소의 내용은 이 표준 범위 밖에서 제공되는 정의에 따라 해석되어야 한다. 만약 이와 같은 포맷에 대한 정의가 별다르게 언급되지 않는다면, 익명성(anonymity), 의사익명성(pseudonymity) 및 주장과 의존하는 기관들에 대한 식별자들의 지속성(persistence) 등은 구현에 따라 고유하게 결정된다(implementation-specific).

SPProvidedID [Optional]

만약 엔티티 이름 식별자가 요소의 내용에서 주어진 주 이름 식별자(primary name identifier)와 다르다면, 서비스 제공자 또는 제공자 연합기관이 이 엔티티에 대하여 설정한 이름 식별자. 이 속성은 서비스 제공자에 의해 사용중인, 이미 존재하는 식별자들이 SAML 에서 통합되어 사용될 수 있는 수단을 제공한다. 예를 들어, 기존의 식별자들은 3.6 절에서 정의된 이름 식별자 관리 프로토콜을 사용하여 엔티티에 부착(attach)될 수 있다.

이들 속성들의 내용(또는 생략)에 대한 추가적인 규칙들은 이 타입을 사용하는 요소와 특정한 Format 정의들에서 정의될 수 있다. 만약 식별자의 타입 정의가 명시적으로 그들의 사용과 의미를 정의하고 있지 않다면, NameQualifier 와 SPNameQualifier 속성들은 생략해야 한다.

다음 스키마 조각은 **NameIDType** 복합 타입을 정의한다.

```
<complexType name="NameIDType">
  <simpleContent>
    <extension base="string">
      <attributeGroup ref="saml:IDNameQualifiers"/>
      <attribute name="Format" type="anyURI" use="optional"/>
      <attribute name="SPProvidedID" type="string" use="optional"/>
    </extension>
  </simpleContent>
</complexType>
```


2.2.3. <NameID> 요소

<NameID> 요소는 **NameIDType** 타입(2.2.2 절 참조)이며, <Subject>와 <SubjectConfirmation> 요소들과 같은 다양한 SAML 주장 구조와 다양한 프로토콜 메시지들(3 장 참조)에서 사용된다.

다음 스키마 조각은 <NameID> 요소를 정의한다.

```
<element name="NameID" type="saml:NameIDType"/>
```

2.2.4. <EncryptedID> 요소

<EncryptedID> 요소는 **EncryptedElementType** 타입이며 W3C Encryption 에서 정의된 것처럼, 암호화되지 않은 식별자 요소의 내용을 암호화하여 지니고 있다(carry). <EncryptedID> 요소는 다음 요소들을 포함한다:

<xenc:EncryptedData> [Required]

W3C Encryption 에서 정의된 것처럼, 암호화된 내용과 관계된 암호화 세부사항을 나타냄. Type 속성은 존재하여야 하며, 만약 존재하면 반드시 <http://www.w3.org/2001/04/xmlenc#Element> 를 값으로 가져야만 한다. 암호화된 내용은 **NameIDType** 또는 **AssertionType** 을 타입으로 가지는 요소를 포함해야만 하거나 또는 **BaseIDAbstractType**, **NameIDType** 또는 **AssertionType** 으로부터 유도된 타입을 가지는 요소를 포함해야만 한다.

<xenc:EncryptedKey> [Zero or More]

W3C Encryption 에서 정의된 것처럼, 포장된 복호화 키들(wrapped decryption keys)을 나타냄. 각각의 포장된 키는 그것이 어떤 엔티티를 대상으로 암호화 되었는지를 나타내도록 엔티티를 명기하는 Recipient 속성을 포함해야 한다. Recipient 속성은 5 장에서 정의된 것처럼, SAML 시스템 엔티티에 대한 URI 식별자이어야 한다.

평문 값(plain-text value)이 중개자를 통해 전달될 때, 프라이버시 보호 메커니즘으로 암호화된 식별자들이 이용된다. 이와 같이, 암호문은 어떠한 암호화 연산에 대해서 유일해야만 한다. 이와 같은 이슈에 대한 더 많은 정보는 W3C XML Encryption 에서 6.3 절을 참조한다.

전체 주장이 이 요소로 암호화되어 식별자로서 사용될 수 있다. 이와 같은 경우, 암호화된 주장의 <Subject> 요소는 이 요소를 둘러싸는(enclosing) 주장의 주체에 대한 식별자를 제공한다. 따라서, 만약 식별하는 주장이 유효하지 않으면, 둘러싸는 주장도 유효하지 않다.

다음 스키마 조각은 <EncryptedID> 요소와 **EncryptedElementType** 복합 타입을 정의한다.

```

<complexType name="EncryptedElementType">
    <sequence>
        <element ref="xenc:EncryptedData"/>
        <element ref="xenc:EncryptedKey" minOccurs="0"
            maxOccurs="unbounded"/>
    </sequence>
</complexType>

<element name="EncryptedID" type="saml:EncryptedElementType"/>

```

2.2.5. <Issuer> 요소

NameIDType 타입을 가지는 <Issuer> 요소는 SAML 주장 또는 프로토콜 메시지의 발급자에 대한 정보를 제공한다. 이 요소는 발급자의 이름을 나타내기 위해 문자열을 사용하도록 요구하지만 여러 가지 설명적인 데이터를 허용하기도 한다(2.2.2 절 참조).

이 요소의 타입에 대한 일반적인 규칙을 오버라이드(override) 하기 때문에, 만약 어떠한 Format 값도 제공되지 않는다면, urn:oasis:names:tc:SAML:2.0:nameid-format:entity (8.3.6 절을 참조) 값이 효과를 갖게 된다.

다음 스키마 조각은 <Issuer> 요소를 정의한다.

```
<element name="Issuer" type="saml:NameIDType"/>
```

2.3. 주장(Assertions)

이번 절은 주장 정보를 포함하거나 또는 이미 존재하는 주장을 참조하는 수단을 제공하는 SAML 구조를 정의한다.

2.3.1. <AssertionIDRef> 요소

<AssertionIDRef> 요소는 그것의 유일 식별자를 통해 SAML 주장을 참조하게 한다. 주장을 발급한 특정 기관(specific authority) 또는 주장을 그것으로부터 얻어올 수 있는 특정 기관은 참조의 일부로써 명기되지는 않는다. 대응되는 주장을 요구하기 위해 이와 같은 참조를 사용하는 프로토콜 요소에 대한 자세한 설명은 3.3 절을 참조한다.

다음 스키마 조각은 <AssertionIDRef> 요소를 정의한다.

```
<element name="AssertionIDRef" type="NCName"/>
```

2.3.2. <AssertionURIRef> 요소

<AssertionURIRef> 요소는 URI 참조를 통해 SAML 주장을 참조하게 한다. URI 참조는 그것에 고유한(specific) 방식으로 대응되는 주장을 검색하는데 사용될 수 있다. 이 요소가 주장을 검색하기 위해 프로토콜 바인딩에서 어떻게 사용되는지에 대한 자세한 정보 바인딩 표준을 참조한다.

다음 스키마 조각은 <AssertionURIRef> 요소를 정의한다.

```
<element name="AssertionURIRef" type="anyURI"/>
```

2.3.3. <Assertion> 요소

<Assertion> 요소는 **AssertionType** 복합 타입이다. 이 타입은 모든 주장들에 공통적인 기본 정보를 명기하며 다음 요소들과 속성들을 포함한다.

Version [Required]

이 주장의 버전. 이 표준에서 정의된 SAML 의 버전을 위한 식별자는 “2.0” 이다. SAML 버전에 대한 내용은 4 장에서 논의된다.

ID [Required]

이 주장의 식별자. 이 요소의 타입은 xs:ID 이고 식별자 유일성을 위해 반드시 1.6.4 절에 명기된 요구조건을 따라야만 한다.

IssueInstant [Required]

이 주장의 발급 시각. 1.6.3 절에서 설명된 것처럼 UTC 형식으로 표현된다.

<Issuer> [Required]

주장(assertion) 안에 여러 주장(claim)들을 만드는 SAML 기관. 주장의 생성이 의도된 의지하는 기관들에게 발급자(issuer)가 모호하지 않아야 한다.

이 표준은 이 요소로 나타내지는 엔티티와 (만약 있다면) 주장을 서명하는 서명자 사이에 특정한 관계를 정의하지 않는다. 주장을 사용하는 의지하는 기관이나 또는 특정한 프로파일에서 지워지는 이와 같은 어떠한 요구사항들도 어플리케이션에 따라 결정된다.

<ds:Signature> [Optional]

아래 부분과 5 장에서 설명하는 것과 같이, 주장의 무결성을 보호하고 발급자를 인증하는 XML 서명

<Subject> [Optional]

주장 내의 문장(들)의 주체

<Conditions> [Optional]

주장의 유효성을 평가(assess)할 때 그리고/또는 주장을 사용할 때, 평가(evaluated) 되어야만 하는 조건들. 조건(conditions)을 어떻게 평가하는지에 대한 추가적인 정보는 2.5 절을 참조한다.

<Advice> [Optional]

어떠한 상황에서 주장의 처리를 지원하기 위해 제공되는 주장과 관련된 추가정보. 그러나 이 요소를 이해하지 못하거나 사용하지 않으려는 응용에서는 이 요소가 무시될 수 있다.

주장은 영 또는 하나 이상의 다음 문장 요소들을 포함할 수 있다.

<Statement>

확장 스키마로 정의된 타입의 문장. xsi:type 속성이 실질적인 문장 타입을 가리키기 위해 반드시 사용되어야만 한다.

<AuthenticationStatement>

인증 문장.

<AuthorizationDecisionStatement>

인가결정 문장.

<AttributeStatement>

속성 문장.

문장을 가지고 있지 않은 주장은 반드시 <Subject> 요소를 포함해야만 한다. 이와 같은 주장은 SAML 방식을 사용하여 참조되거나 또는 확인(confirm)될 수 있는 방식으로 사용자(principal)을 식별한다. 그러나 그 사용자와 연관된 더 이상의 정보를 보장하지는 않는다.

만약 주장이 문장을 가지고 있고, <Subject>가 존재하면, <Subject>는 주장 내에 있는 모든 문장들의 주체를 식별한다. 만약 <Subject>가 생략되면, 주장 내의 문장들은 어플리케이션 또는 프로파일에서 정하는 고유한 방식으로 식별되는 주체 또는 주체들에게 적용된다. SAML 은 이와 같은 문장을 정의하고 있지 않으며 주체가 없는 주장은 이 표준에서 어떠한 정의된 의미를 가지지 않는다.

특정 프로토콜 또는 프로파일의 요구조건에 따라, SAML 주장의 발급자는 자주 인증될 필요가 있을 수 있으며 무결성 보호가 요구될 수도 있다. 인증과 메시지 무결성은 주장을 배달하는 동안 사용되는 프로토콜 바인딩에 의해 제공되는 메커니즘이 제공될 수 있다(SAML 2.0 바인딩 표준 참조). SAML 주장은 서명될 수 있으며 이것은 발급자의 인증과 무결성 보호 두 가지 기능 모두를 제공한다.

이와 같은 서명이 사용되면, <ds:Signature> 요소가 반드시 존재해야만 하고 의지하는 측은 반드시 서명이 W3C XML Signature 에 따라 유효하다는 것(즉, 주장이 변조(tamper)되지 않았다는 사실)을 확인해야만 한다. 만약 이것이 유효하면, 의지하는 측은 발급자의 신원과 적절성을 판단하기 위해 서명을 평가해야 하고 이 표준에 따라서 주장의 처리를 계속할 수 있다.

서명되던 또는 서명되지 않던 간에, (주체, 조건, 기타 사항이 동일하다면) 단일한 주장에 여러 문장을 포함하는 것은 이들 문장들을 개별적으로 포함하는 주장 집합과 의미적으로 동일하다는 것에 주의해야 한다.

다음 스키마 조각은 <Assertion> 요소와 **AssertionType** 복합 타입을 정의한다.

```
<element name="Assertion" type="saml:AssertionType"/>
  <complexType name="AssertionType">
    <sequence>
      <element ref="saml:Issuer"/>
      <element ref="ds:Signature" minOccurs="0"/>
      <element ref="saml:Subject" minOccurs="0"/>
      <element ref="saml:Conditions" minOccurs="0"/>
      <element ref="saml:Advice" minOccurs="0"/>
      <choice minOccurs="0" maxOccurs="unbounded">
        <element ref="saml:Statement"/>
        <element ref="saml:AuthnStatement"/>
        <element ref="saml:AuthzDecisionStatement"/>
        <element ref="saml:AttributeStatement"/>
      </choice>
    </sequence>
    <attribute name="Version" type="string" use="required"/>
    <attribute name="ID" type="ID" use="required"/>
    <attribute name="IssueInstant" type="dateTime" use="required"/>
  </complexType>
```

2.3.4. <EncryptedAssertion> 요소

<EncryptedAssertion> 요소는 W3C Encryption 에서 정의된 것처럼, 주장을 암호화된 방식으로 나타낸다. <EncryptedAssertion> 요소는 다음과 같은 요소들을 포함한다.

<xenc:EncryptedData> [Required]

W3C Encryption 에서 정의된 것처럼, 암호화된 내용과 관계된 암호화 세부사항을 나타냄. Type 속성은 존재해야 하며 만약 존재하면 반드시 <http://www.w3.org/2001/04/xmlenc#Element> 를 값으로 가져야만 한다. 암호화된 내용은 AssertionType 을 타입의 요소 또는 AssertionType 으로부터 유도된 타입의 요소를 포함해야만 한다.

<xenc:EncryptedKey> [Zero or More]

W3C Encryption 에서 정의된 것처럼, 포장된 복호화 키들(wrapped decryption keys)을 나타냄. 각각의 포장된 키는 그것이 어떤 엔티티를 대상으로 암호화 되었는지를 나타내도록 엔티티를 기술하는 Recipient 속성을 포함해야 한다. Recipient 속성은 8.3.6 절에서 정의된 것처럼, SAML 시스템 엔티티에 대한 URI 식별자이어야 한다.

평문 값(plain-text value)이 중개자를 통해 전달될 때, 프라이버시 보호 메커니즘으로 암호화된 주장들이 이용된다.

다음 스키마 조각은 <EncryptedAssertion> 요소를 정의한다.

```
<element name="EncryptedAssertion" type="saml:EncryptedElementType"/>
```

2.4. 주체들(Subjects)

이번 절은 주장의 주체를 설명하는데 사용되는 SAML 구조를 정의한다.

선택적인 <Subject> 요소는 주장에 포함되는 (영 또는 하나 이상의) 모든 문장들의 주체인 사용자(principal)를 기술한다. 이 요소는 식별자 또는 하나 이상의 주체 확인(confirmations)을 포함하거나 또는 두 가지 모두를 포함한다.

<BaseID>, <NameID> 또는 <EncryptedID> [Optional]

주체를 식별한다.

<SubjectConfirmation> [Zero or More]

주체가 확인(confirm) 되도록 해 주는 정보. 만약 하나 이상의 주체 확인이 제공되는 경우, 그들 중에 어느 하나라도 만족하면, 주장을 적용하기 위하여 주체를 확인하는 것이 충족되었다고 판단할 수 있다.

의지하는 기관이 주장을 처리할 때 검증할 수 있는 식별자와 영 또는 그 이상의 주체 확인을 <Subject> 요소는 포함할 수 있다. 만약 포함된 주체 확인들 중에서 어느 하나가 검증되면, 의지하는 기관은 주장을 제출한 엔티티가, 보장하는 기관이 이름 식별자로 식별되는 사용자(principal)와 연관시키고 주장에서 문장들과 연관시키는 엔티티인 것으로 간주할 수 있다. 증명하는(attesting) 엔티티와 실제 주체는 동일한 엔티티일 수도 있고 아닐 수도 있다.

<Subject> 요소는 하나 이상의 사용자(principal)을 식별하지 않아야 한다.

다음 스키마 조각은 <Subject> 요소와 **SubjectType** 복합 타입을 정의한다.

```
<element name="Subject" type="saml:SubjectType"/>
```

```
<complexType name="SubjectType">
```

```
<choice>
```

```
<sequence>
```

```
<choice>
```

```
<element ref="saml:BaseID"/>
```

```
<element ref="saml:NameID"/>
```

```
<element ref="saml:EncryptedID"/>
```

```
</choice>
```

```
<element ref="saml:SubjectConfirmation" minOccurs="0"
```

```
maxOccurs="unbounded"/>
```

```
</sequence>
```

```
<element ref="saml:SubjectConfirmation" maxOccurs="unbounded"/>
```

```
</choice>
```

```
</complexType>
```

2.4.1. <SubjectConfirmation> 요소

<SubjectConfirmation> 요소는 의지하는 기관이 주장의 주체가 자신이 통신하고 있는 기관과 일치하는지를 검증하는 수단을 제공한다. 이것은 다음 속성들과 요소들을 포함한다.

Method [Required]

주체를 확인하는데 사용되는 프로토콜 또는 메커니즘을 식별하는 URI 참조. SAML 에서 정의된 확인 방법들을 식별하는 URI 참조들은 현재 SAML 2.0 프로파일 표준에 정의되어

있다. 추가적인 방법들은 새로운 URI 들과 프로파일들을 정의하거나 또는 개인적인 협약(private agreement)을 통해 추가될 수 있다.

<BaseID>, <NameID> 또는 <EncryptedID> [Optional]

이 요소를 둘러싸는 주체 확인 요구사항을 충족시킬 것으로 기대되는 엔티티를 식별한다.

<SubjectConfirmationData> [Optional]

특수한 확인 방법에 의해 사용되는 추가적인 확인 정보. 예를 들어, 이 요소의 전형적인 내용은 W3C XML Signature 에서 정의된 것과 같은 <ds:KeyInfo> 요소가 될 수 있으며, 이것은 암호화 키를 식별한다(2.4.1.3 절 또한 참조). 특정 확인 방법들은 <SubjectConfirmationData> 요소에 나타날 수 있는 요소들, 속성들 또는 내용을 설명하는 스키마 타입을 정의할 수 있다.

다음 스키마 조각은 <SubjectConfirmation> 요소와 **SubjectConfirmationType** 복합 타입을 정의한다.

```
<element name="SubjectConfirmation" type="saml:SubjectConfirmationType"/>
```

```
<complexType name="SubjectConfirmationType">
```

```
<sequence>
```

```
<choice minOccurs="0">
```

```
<element ref="saml:BaseID"/>
```

```
<element ref="saml:NameID"/>
```

```
<element ref="saml:EncryptedID"/>
```

```
</choice>
```

```
<element ref="saml:SubjectConfirmationData" minOccurs="0"/>
```

```
</sequence>
```

```
<attribute name="Method" type="anyURI" use="required"/>
```

```
</complexType>
```


2.4.2. <SubjectConfirmationData> 요소

<SubjectConfirmationData> 요소는 **SubjectConfirmationDataType** 복합 타입을 가진다. 이 요소는 주체가 확인되도록 하는 추가적인 데이터를 기술하거나 또는 주체 확인 동작이 발생하는 환경을 제한한다. 주체 확인은 의지하는 기관이 주장을 제출하는 엔티티(즉, 증명하는 엔티티(attesting entity))와 주장 내 주장(assertion's claim)의 주체간의 관계를 검증하려 할 때 발생한다. 이것은 어떠한 방식에도 적용될 수 있는 다음 선택적인 속성들을 포함한다.

NotBefore [Optional]

시각 값. 주체가 이 값 이전에는 확인될 수 없는 시각을 나타냄. 시각 값은 1.6.3 절에서 설명된 것처럼 UTC 형식으로 인코딩된다.

NotOnOrAfter [Optional]

시각 값. 주체가 이 시각 또는 이 시각 이후에는 더 이상 확인될 수 없는 시각을 나타냄. 시각 값은 1.6.3 절에서 설명된 것처럼 UTC 형식으로 인코딩된다.

Recipient [Optional]

증명하는 엔티티가 주장을 제출할 수 있는 엔티티나 위치를 기술하는 URI. 예를 들어, 이 속성은 중개자(intermediary)가 주장을 다른 장소로 전달하는(redirect) 것을 방지하기 위해 주장이 특정한 네트워크 엔드포인트(endpoint)으로 배달되어야만 한다는 것을 가리킬 수 있다.

InResponseTo [Optional]

증명하는 엔티티가 주장을 응답으로 전달하게 하는 원래의 SAML 프로토콜 메시지의 ID. 예를 들어, 이 속성은 주장과 주장의 제출을 유발시키는 SAML 요청을 서로 연관시키는데 사용될 수 있다.

Address [Optional]

증명하는 엔티티가 주장을 어디에서 제출할 수 있는지를 나타내는 네트워크 주소/위치. 예를 들어, 이 속성은 주장을 특정한 클라이언트 주소들에 바인딩할 수 있도록 하여, 공격자가 주장을 쉽게 훔쳐 다른 위치로부터 주장을 제출하는 것을 방지할 수 있다. IPv4 주소들은 일반적인 점-10 진(dotted-decimal) 형식(예, "1.2.3.4")으로 표현되어야 한다. IPv6 주소들은 IETF RFC 3513 의 2.2 절에서 정의된 것처럼 (예, "FEDC:BA98:7654:3210:FEDC:BA98:7654:3210") 표현되어야 한다.

임의의 속성들 (Arbitrary attributes)

이 복합 타입은 <xs:anyAttribute> 확장점을 사용하여, 명시적으로 스키마를 확장할 필요 없이, 임의의 네임스페이스로 제한된(namespace-qualified) XML 속성들이 <SubjectConfirmationData> 구조에 추가하는 것을 허용한다. 추가적인 확인 관련 정보를 제공할 필요가 있을 때마다, 이 속성은 추가적인 필드들이 추가될 수 있도록 허용한다. SAML 확장들은 지역(local) (non-namespace-qualified) XML 속성들 또는 SAML 에서 정의된 네임스페이스로 제한되는 XML 속성들이 **SubjectConfirmationDataType** 복합 타입 또는 그것으로 유도된 타입에 추가되지 않아야만 한다. 이와 같은 속성들은 SAML 자체에 대한 미래의 유지보수와 개선을 위해 예약되어 있다.

임의의 요소들 (Arbitrary elements)

이 복합 타입은 명시적으로 스키마를 확장할 필요 없이, 임의의 XML 요소들이 추가될 수 있도록 하기 위해 <xs:any> 확장점을 사용한다. 추가적인 확인 관련 정보를 제공할 필요가 있을 때마다, 이 요소는 추가적인 요소들이 추가될 수 있도록 허용한다.

특정한 확인 방법들과 그러한 방법들을 이용하는 프로파일들은 이 복합 타입내에서 정의된 하나 또는 그 이상의 속성들을 사용할 것은 요구할 수 있다. 이들 속성들이 (그리고 일반적으로 주체 확인이) 어떻게 사용될 수 있는지에 대한 예에 대해서는 프로파일 표준을 참조한다.

선택적인 NotBefore 와 NotOnOrAfter 속성들에 의해 명기된 시각 구간은, 만약 그것이 존재하면, <Conditions> 요소의 NotBefore 와 NotOnOrAfter 속성들에 의해 명기된 전체 주장 유효기간 내에 있어야 한다는 사실에 주의해야 한다. 만약 두가지 속성들이 모두 존재하면, NotBefore 의 값은 반드시 NotOnOrAfter 값보다 작아야만 한다.

다음 스키마 조각은 <SubjectConfirmationData> 요소와 **SubjectConfirmationDataType** 복합 타입을 정의한다.

```
<element name="SubjectConfirmationData"
  type="saml:SubjectConfirmationDataType"/>

<complexType name="SubjectConfirmationDataType" mixed="true">
  <complexContent>
    <restriction base="anyType">
      <sequence>
        <any namespace="##any" processContents="lax" minOccurs="0"
          maxOccurs="unbounded"/>
      </sequence>
      <attribute name="NotBefore" type="dateTime" use="optional"/>
    </restriction>
  </complexContent>
</complexType>
```

```

<attribute name="NotOnOrAfter" type="dateTime" use="optional"/>
  <attribute name="Recipient" type="anyURI" use="optional"/>
<attribute name="InResponseTo" type="NCName" use="optional"/>
  <attribute name="Address" type="string" use="optional"/>
  <anyAttribute namespace="##other" processContents="lax"/>
</restriction>
</complexContent>
</complexType>

```

2.4.3. KeyInfoConfirmationDataType 복합 타입

KeyInfoConfirmationDataType 복합 타입은 <SubjectConfirmationData> 요소가 증명하는 엔티티를 인증하기 위하여 어떤 방식으로 사용되는 암호 키들을 식별하는 하나 또는 그 이상의 <ds:KeyInfo> 요소들을 포함하도록 제한한다. 특정한 확인 방법은 확인 데이터가 사용될 수 있는 정확한 메커니즘을 정의해야만 한다.

SubjectConfirmationDataType 복합 타입에서 정의된 선택적인 속성들 또한 나타날 수 있다.

그것의 확인 데이터를 <ds:KeyInfo> 요소로 정의하는 어떠한 확인 방법도 이 복합 타입 또는 이것으로부터 유도된 타입을 사용해야 한다.

W3C Encryption 에 따라서, 각각의 <ds:KeyInfo> 요소는 반드시 단일한 암호화 키를 식별해야만 한다. 사용자(principal)가 자신을 다른 의지하는 기관들에게 확인시키기 위해 다른 키들을 사용하는 때와 같은 경우, 다중 키들은 분리된 <ds:KeyInfo> 요소들로 식별될 수 있다.

다음 스키마 조각은 **KeyInfoConfirmationDataType** 복합 타입을 정의한다.

```

<complexType name="KeyInfoConfirmationDataType" mixed="false">
  <complexContent>
    <restriction base="saml:SubjectConfirmationDataType">
      <sequence>
        <element ref="ds:KeyInfo" maxOccurs="unbounded"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>

```

2.4.4. 키-확인된 <Subject>의 예 (Example of a Key-Confirmed <Subject>)

어떠한 방식으로 다양한 요소들과 타입들이 서로 결합되는지 그 방식을 설명하기 위해, 아래에 이름 식별자를 포함하는 <Subject> 요소와 키 소유 증명(proof of possession of a key)을 기반으로 주체를 확인하는 예를 기술한다. 확인 데이터 문법을 <ds:KeyInfo> 요소인 것으로 식별하기 위해 **KeyInformationDataType** 을 사용한 것에 주의해야 한다.

```

<Subject>
  <NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
    scott@example.org
  </NameID>
  <SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-
    key">
    <SubjectConfirmationData xsi:type="saml:KeyInfoConfirmationDataType">
      <ds:KeyInfo>
        <ds:KeyName>Scott's Key</ds:KeyName>
      </ds:KeyInfo>
    </SubjectConfirmationData>
  </SubjectConfirmation>
</Subject>

```

2.5. 조건들(Conditions)

이번 절은 SAML 주장들에 대한 수용가능한 사용(acceptable use)에 제약들 두는 SAML 구조를 정의한다. <Conditions> 요소는 다음 요소들과 속성들을 포함할 수 있다.

NotBefore [Optional]

주장의 유효기간 중에 가장 빠른 시각인 유효기간 시작 시점을 기술한다. 시각 값은 1.6.3 절에서 설명된 것처럼 UTC 형식으로 인코딩된다.

NotOnOrAfter [Optional]

주장이 만료되는 시점을 기술한다. 시각 값은 1.6.3 절에서 설명된 것처럼 UTC 형식으로 인코딩된다.

<Condition> [Any Number]

확장 스키마에서 정의된 타입의 조건. xsi:type 속성이 실질적인 조건 타입을 가리키기 위해 사용되어야만 한다.

<AudienceRestriction> [Any Number]

주장이 특정한 청중들에게만 전달되도록 기술한다.

<OneTimeUse> [Optional]

주장이 즉시 사용되어야 하며 미래의 사용을 위해 보유(retain)되지 않아야만 한다는 사실을 기술한다. 비록 스키마에서는 여러 번 이 요소가 나타나는 것을 허용하지만, 이 요소는 <Conditions> 요소에서 많아야 한번 나타나야만 한다.

<ProxyRestriction> [Optional]

보장하는 기관(assertion party)이 의지하는 기관들에게 가하는 제약을 명기한다. 이 경우, 의지하는 기관들은 순차적으로 자신들이 보장하는 기관으로서 동작하기를 원하며 보장하는 기관이 자신들에게 전달한 원래의 주장에 포함된 정보를 근거로하여 새로운 주장들을 발급한다. 비록 스키마에서는 여러 번 이 요소가 나타나는 것을 허용하지만, 이 요소는 <Conditions> 요소에서 많아야 한번 나타나야만 한다.

xsi:type 속성을 사용하면 주장이 (OneTimeUseType 과 같은) SAML 에서 정의된 **ConditionsType** 의 하부타입(subtype)의 인스턴스(instance)를 하나 이상 포함하는 것이 허용되기 때문에, 스키마는 명시적으로 특정한 조건들이 몇 번 포함될 수 있는지를 제약하지 않는다. 특정한 타입의 조건은 위에서 보여진 것처럼 이와 같은 사용에 대한 제약을 정의할 수 있다.

다음 스키마 조각은 <Conditions> 요소와 **ConditionsType** 복합 타입을 정의한다.

```
<element name="Conditions" type="saml:ConditionsType"/>
```

```
<complexType name="ConditionsType">
```

```
<choice minOccurs="0" maxOccurs="unbounded">
```

```
<element ref="saml:Condition"/>
```

```
<element ref="saml:AudienceRestriction"/>
```

```
<element ref="saml:OneTimeUse"/>
```

```
<element ref="saml:ProxyRestriction"/>
```

```
</choice>
```

```
<attribute name="NotBefore" type="dateTime" use="optional"/>
```

```
<attribute name="NotOnOrAfter" type="dateTime" use="optional"/>
```

```
</complexType>
```

2.5.1. 일반적인 처리 규칙들

만약 주장이 <Conditions> 요소를 포함한다면, 주장의 유효성은 제공되는 하부 요소들과 속성들에 따라 결정되는데, 아래 보여지는 순서대로 다음 규칙들을 사용하게 된다.

조건 유효 상태로 **Valid** 를 갖는 주장도 XML 문법에 적합하지 않거나(not well-formed) 또는 유효하지 않거나, 신뢰되는 SAML 기관에서 발급되지 않거나 또는 신뢰되는 수단으로 인증되지 않는 것과 같은 이유로 신뢰되지 않거나 또는 유효하지 않을 수 있다.

일부 조건들은 조건을 포함하는 주장의 유효성에 직접적으로 영향을 주지 않을 수 있지만, 그러나 주장의 사용에 대해서 의지하는 기관이 행동하는 것에 제약을 가할 수는 있다.

1. 만약 <Conditions> 요소에 하부 요소들과 속성들이 존재하지 않는다면, 주장은 조건 처리에 있어서는 유효한 것으로 간주된다.
2. 만약 <Conditions> 요소의 어떠한 하부 요소들과 속성이 유효하지 않다고 판단되면, 주장은 유효하지 않은 것으로 간주된다.
3. 만약 <Conditions> 요소의 어떠한 하부 요소들과 속성이 평가될(evaluated) 수 없거나 또는 이해되지 않는 요소가 존재하면, 주장의 유효성은 판단할 수 없으며 불확정한(Indeterminate) 것으로 간주된다.
4. 만약 <Conditions> 요소의 모든 하부 요소들과 속성들이 유효한 것으로 판단되면, 주장은 조건 처리에 대해서는 유효한 것으로 간주된다.

어떠한 규칙이 적용되면, 그 규칙은 조건 처리를 종료시킨다; 따라서, 주장이 유효하지 않다는(Invalid) 판단이 불확정성(Indeterminate) 보다 우선 순위가 높다.

주장이 유효하지 않거나 또는 불확정한 것으로 판단된다면, 마치 주장이 잘못된 형태를 띄고 있거나(malformed) 또는 사용할 수 없게 된 것처럼, 그 주장은 (그것이 처리되는 문맥이나 프로파일 내에서) 의지하는 기관에 의해 거절되어야만 한다.

2.5.2. NotBefore 와 NotOnOrAfter 속성들

NotBefore 와 NotOnOrAfter 속성들은 사용중인 프로파일 문맥 내에서 주장의 유효성에 시각 제약(time limit)을 명기한다. 이들이 주장에 존재하는 문장들이 유효기간 내내 올바르게 정확할 것이라는 것을 보장하지는 않는다.

NotBefore 속성은 유효기간이 시작되는 시각을 기술한다. NotOnOrAfter 속성은 유효기간이 종료되는 시각을 기술한다.

만약 NotBefore 또는 NotOnOrAfter 에 대한 값이 생략되면, 그 값이 명기되지 않은 것으로 간주된다. 만약 NotBefore 값이 명기되지 않는다면 (그리고 만약 제공되는 모든 다른 조건들이 유효한 것으로 평가되면), 그러면 NotOnOrAfter 속성에 의해 명기된 시각 이전의 어떠한 시각에서도 주장은 조건들에 대하여 유효한 것이 된다. 만약 NotOnOrAfter 값이 명기되지 않는다면 (그리고 만약 제공되는 모든 다른 조건들이 유효한 것으로 평가되면), 그러면 NotBefore 속성에 의해 기술된 시각 이후에 존재하는 어떠한 시각에서도 주장은 조건들에 대하여 유효한 것이 된다. 만약 두 속성이 모두 명기되지 않는다면 (그리고 만약 제공되는 모든 다른 조건들이 유효한 것으로 평가되면), 주장은 어떠한 시각에서도 조건들에 대하여 유효한 것이다.

2.5.3. <Condition> 요소

<Condition> 요소는 새로운 조건들을 위한 확장점으로 역할을 수행한다. 이것의 타입인 **ConditionAbstractType** 복합 타입은 추상적이며 따라서 유도타입의 기본(base)으로만 사용될 수 있다.

다음 스키마 조각은 <Condition> 요소와 **ConditionAbstractType** 복합 타입을 정의한다.

```
<element name="Condition" type="saml:ConditionAbstractType"/>
```

```
<complexType name="ConditionAbstractType" abstract="true"/>
```

2.5.4. <AudienceRestriction>과 <Audience> 요소

<AudienceRestriction> 요소는 주장이 <Audience> 요소들에 의해 식별되는 하나 또는 그 이상의 특수한 청중들에게 전달되도록 명기한다. 비록 주장에서 명기된 청중들 밖에 있는 SAML 을 의지하는 기관이 주장으로부터 어떠한 결론을 도출할 수는 있지만, SAML 에서 보장하는 기관은 청중 밖에 있는 의지하는 기관에게 정확성과 신뢰도에 대한 명시적인 어떠한 표현도 하지 않는다. 이것은 다음 요소를 포함한다.

<Audience>

의도된 청중을 식별하는 URI 참조. URI 참조는 청중 멤버십의 항목과 조건(terms and conditions)을 설명하는 문서를 식별할 수 있다. 이것은 또한 시스템 엔티티를 설명하는 SAML 이름 식별자로부터 유일한 식별자 URI 를 포함할 수도 있다.

SAML 을 의지하는 기관이 명기된 하나 또는 그 이상의 청중들의 멤버인 것은 청중 제약 조건이 유효한 것으로 평가되기 위한 필요충분 조건이다.

SAML 에서 보장하는 기관은 주장을 입수한 다른 기관이 주장에서 제공하는 정보를 기반으로 어떠한 행동을 취하는 것을 방지할 수는 없다. 그러나, <AudienceRestriction> 요소는 SAML 에서 주장하는 기관이 이와 같은 기관에서 기계 또는 인간이 가독할(machine- and human-readable) 수 있는 형태로 명시적으로 어떠한 보장도 할 수 없다는 것을 서술할 수 있는 것을 허용한다. 비록 법원이 이와 같은 보증 배제(warranty exclusion)를 모든 환경에서 지지하리라는 것을 보증할 수는 없지만, 보증 배제를 지지할 확률은 꽤 높아진다.

다중 <AudienceRestriction> 요소들이 단일한 주장에 포함될 수 있으며 각각은 서로 독립적으로 평가되어야만 한다. 이 요구사항과 앞의 정의가 주는 효과는, 주어진 조건 내에서, 청중들은 분리(disjunction, “OR”)을 형성하고 반면에 다중 조건들은 결합(conjunction, “AND”)을 형성한다는 것이다.

다음 스키마 조각은 <AudienceRestriction> 요소와 **AudienceRestrictionType** 복합 타입을 정의한다.


```

<element name="AudienceRestriction" type="saml:AudienceRestrictionType"/>

<complexType name="AudienceRestrictionType">
  <complexContent>
    <extension base="saml:ConditionAbstractType">
      <sequence>
        <element ref="saml:Audience" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<element name="Audience" type="anyURI"/>

```

2.5.5. <OneTimeUse> 요소

일반적으로, 의지하는 기관들은 재사용을 위하여 주장을 보유하거나 또는 주장들이 다른 형태로 포함하는 정보를 보유하기로 선택할지 모른다. <OneTimeUse> 조건 요소는 한 기관이 주장 내 정보가 매우 자주 변화할 것 같으니 각각의 사용시마다 최신의 정보(fresh information)를 얻어와야 한다는 사실을 가리키는 것을 허용한다. 하루의 시각에 대한 함수로 접근 제어를 명기하는 경우, <AuthzDecisionStatement>를 포함하는 주장이 하나의 예가 될 것이다.

만약 분산 환경에서 시스템 클록들(system clocks)이 정밀하게 동기화(synchronize)될 수 있다면, 이러한 요구사항은 유효기간 구간을 주의 깊게 사용함으로써 만족될 수 있을 것이다. 그러나, 시스템간에 어느 정도의 시각 뒀(clock skew)은 항상 발생할 것이고, 가능한 전송 지연과 결합될 것이기 때문에, 주장이 도착하기 전에 이미 유효기간이 종료되는 실질적인 위험 없이, 발급자가 주장의 수명(lifetime)을 적절하게 제약하는 편리한 방법은 존재하지 않는다.

<OneTimeUse> 요소는 주장이 의지하는 기관에서 즉시 사용되어야 한다는 것과 미래의 사용을 위해 보유되지 않아야만 한다는 사실을 가리킨다. 의지하는 기관들은 항상 자유롭게 모든 사용마다 새로운 주장을 요구할 수 있다. 그러나, 미래의 사용을 위해 주장을 보유하기로 선택한 구현들은 반드시 <OneTimeUse> 요소를 주시해야만 한다. 이 조건은 NotBefore 와 NotOnOrAfter 조건 정보와는 독립적이다.

단일한 사용 제약(use constraint)을 지원하기 위해, 의지하는 기관은 그것이 처리해왔던 주장 중에서 이러한 조건을 포함하는 주장들에 대한 캐쉬(cache)을 유지해야 한다. 이러한 조건을 가진 주장이 처리될 때마다, 동일한 주장이 이전에 수신되어 의지하는 기관에서 처리되지 않았다는 것을 확인하기 위해 캐쉬가 검사되어야 한다.

SAML 기관은 주장의 <Conditions> 요소에 하나 이상의 <OneTimeUse> 요소를 포함하지 않아야만 한다.

<Conditions> 요소의 유효성을 판단하기 위해, <OneTimeUse>는 항상 유효한 것으로 간주된다. 즉, 이 조건은 유효성에는 영향을 미치지 않고 사용에 대한 조건이 된다.

다음 스키마 조각은 <OneTimeUse> 요소와 **OneTimeUseType** 복합 타입을 정의한다.

```
<element name="OneTimeUse" type="saml:OneTimeUseType"/>
```

```
<complexType name="OneTimeUseType">
```

```
  <complexContent>
```

```
    <extension base="saml:ConditionAbstractType"/>
```

```
  </complexContent>
```

```
</complexType>
```

2.5.6. <ProxyRestriction> 요소

보장하는 기관(assertion party)이 의지하는 기관들에게 가할 수 있는 제약을 명기한다. 이 경우, 의지하는 기관들은 차례로 자신들이 보장하는 기관으로서 동작하기를 원하며 보장하는 기관이 자신들에게 전달한 원래의 주장에 포함된 정보를 근거로하여 새로운 주장들을 발급한다. 보장하는 기관으로서 동작하는 의지하는 기관은 이와 같은 조건을 포함하는 주장을 근거로 하여 이 조건에 명기된 제약 사항을 위반하는 주장을 발급하지 않아야만 한다.

<ProxyRestriction> 요소는 다음과 같은 요소들과 속성들을 포함한다.

Count [Optional]

보장하는 기관이 이 주장과 이것을 기초로 하여 최종적으로 발급되는 주장 사이에 존재하도록 허용하는 우회(indirection)의 최대 수를 명기한다.

<Audience> [Zero or More]

보장하는 기관이 이 주장을 기초로 하여 발급되는 새로운 주장들이 어떠한 청중을 대상으로 발급되어야 하는 지를 명기한다.

영의 값을 갖는 Count 는 의지하는 기관이 이 주장을 기초로 하여 또 다른 의지하는 기관에게 주장을 발급하지 않아야만 한다는 사실을 가리킨다. 만약 Count 값이 영보다

크면, 그렇게 발급된 어떠한 주장들도 이 값보다 기껏해야 1 작은 Count 값을 갖는 <ProxyRestriction> 요소를 가져야만 한다.

만약 <Audience> 요소들이 명기되지 않으면, 계속적으로 주장들이 발급될 수 있는 의지하는 기관들에게는 어떠한 청중 제약도 가해지지 않는다. 만약 그렇지 않다면, 이렇게 발급된 주장들은 이전 <ProxyRestriction> 요소에서 존재하는 <Audience> 요소들 중 적어도 하나 이상의 <Audience> 요소를 포함하도록 발급해야만 하고 이전 <ProxyRestriction> 요소에 존재하지 않았던 <Audience> 요소들은 포함되지 않도록 해야만 한다.

SAML 기관은 주장의 <Conditions> 요소 내에 하나 이상의 <ProxyRestriction> 요소를 포함하지 않아야만 한다.

<Conditions> 요소의 유효성 판단을 위해, <ProxyRestriction> 조건은 항상 유효한 것으로 간주된다. 즉, 이 조건은 유효성에는 영향을 미치지 않지만 사용에 대한 하나의 조건이 된다.

다음 스키마 조각은 <ProxyRestriction> 요소와 **ProxyRestrictionType** 복합 타입을 정의한다.

```
<element name="ProxyRestriction" type="saml:ProxyRestrictionType"/>

<complexType name="ProxyRestrictionType">
  <complexContent>
    <extension base="saml:ConditionAbstractType">
      <sequence>
        <element ref="saml:Audience" minOccurs="0"
          maxOccurs="unbounded"/>
      </sequence>
      <attribute name="Count" type="nonNegativeInteger" use="optional"/>
    </extension>
  </complexContent>
</complexType>
```

2.6. 충고(Advice)

이번 절은 보장하는 기관이 의지하는 기관에게 제공하기를 원하는 주장에 대한 추가적인 정보를 포함하는 SAML 구조를 정의한다.

<Advice> 요소는 SAML 기관이 제공하기를 원하는 어떠한 추가적인 정보를 포함한다. 이 정보는 주장의 의미나 또는 유효성에 영향을 미치지 않고 어플리케이션에 의해 무시될 수 있다.

<Advice> 요소는 <Assertion>, <EncryptedAssertion>, <AssertionIDRef>와 <AssertionURIRef> 요소들과 다른 SAML 이 아닌(non-SAML) 네임스페이스에 속하는 네임스페이스 제한된(namespace-qualified) 요소들의 영 또는 그 이상의 혼합을 포함한다.

<Advice 요소>은 다음과 같은 식으로 사용될 수 있다.

- 직접적으로(주장들을 통합함으로써) 또는 간접적으로(지원하는 주장들을 참조함으로써) 주장의 주장들(assertion claims)이 인용되는 것을 지원하는 증거를 포함하기
- 주장의 주장들(assertion claims)의 증명을 서술하기
- 주장에 대한 갱신을 위한 타이밍과 분배점(timing and distribution points)을 명기하기

다음 스키마 조각은 <Advice> 요소와 **AdviceType** 복합 타입을 정의한다.

```
<element name="Advice" type="saml:AdviceType"/>

<complexType name="AdviceType">
  <choice minOccurs="0" maxOccurs="unbounded">
    <element ref="saml:AssertionIDRef"/>
    <element ref="saml:AssertionURIRef"/>
    <element ref="saml:Assertion"/>
    <element ref="saml:EncryptedAssertion"/>
    <any namespace="##other" processContents="lax"/>
  </choice>
</complexType>
```

2.7. 문장(Statements)

모든 SAML 에서 정의된 문장들은 주체와 연관된다. SAML 주장들은 일반적으로 <Subject> 요소로 표현되는 **Subject** 에 대하여 생성된다. 그러나, <Subject> 요소는 선택적이고, 다른 표준과 프로파일들은 주체를 기술하기 않고 또는 다른 방식으로 주체를 기술함으로써, 유사한 문장들을 생성하는데 SAML 주장 구조를 활용할 수 있다. 다음 절은 문장 정보를 포함하는 SAML 구조를 정의한다.

2.7.1. <Statement> 요소

<Statement> 요소는 다른 주장-기반 어플리케이션이 SAML 주장 프레임워크를 재사용하도록 해 주는 확장점이다. SAML 자체는 그것의 핵심 문장을 이 확장점으로부터 유도한다. 이 요소의 타입인 **StatementAbstractType** 복합 타입은 추상적이며 따라서 유도 타입의 기본(base)로만 사용될 수 있다.

다음 스키마 조각은 <Statement> 요소와 **StatementAbstractType** 복합 타입을 정의한다.

```
<element name="Statement" type="saml:StatementAbstractType"/>

<complexType name="StatementAbstractType" abstract="true"/>
```

2.7.2. <AuthnStatement> 요소

<AuthnStatement> 요소는 주장 주체가 특정한 시각에 특정한 수단에 의해 인증되었다는 것을 보장하는 SAML 기관에 의해 생성하는 문장을 설명한다. <AuthnStatement> 요소들을 포함하는 주장들은 주체 요소를 포함해야만 한다.

이 요소는 **AuthnStatementType** 타입이며, 이 타입은 다음 요소들과 속성들을 추가하여 StatementAbstractType 을 확장한다:

주의: <AuthorityBinding> 요소와 이 요소와 대응되는 타입은 SAML 2.0 에서 <AuthnStatement>로부터 제거되었다.

AuthnInstant [Required]

인증과정이 발생한 시각을 명기한다. 시각 값은 1.6.3 절에서 설명된 것처럼 UTC 형식으로 인코딩된다.

SessionIndex [Optional]

주체에 의해 식별되는 사용자(principal)와 인증 기관 사이의 특정한 세션(particular session)의 인덱스를 기술한다.

SessionNotOnOrAfter [Optional]

주체에 의해 식별되는 사용자(principal)와 이 문장을 발급하는 SAML 기관 사이에서 세션이 종료될 것으로 간주되어야만 하는 시각을 명기한다. 시각 값은 1.3.3 절에서 설명된 것처럼 UTC 형식으로 인코딩된다. 이 속성과 주장에서 존재하는 NotOnOrAfter 조건 사이에는 특별히 요구되는 관계가 존재하지는 않는다.

<SubjectLocality> [Optional]

주장 주체가 어느 시스템으로부터 인증이 되었는지를 나타내는 DNS 도메인 이름과 IP 주소 값을 기술한다.

<AuthnContext> [Required]

이 문장을 생성하는 인증 사건(event)를 포함하여 인증 기관(authentication authority)에 의해 사용된 문맥, 인증 문맥 클래스 참조를 포함하거나 인증 문맥 선언 또는 선언 참조를 포함하거나 또는 둘 다를 포함한다. 인증 문맥 정보에 대한 자세한 설명은 인증 문맥 표준을 참조한다.

일반적으로, 어떠한 문자열도 SessionIndex 값으로 사용될 수 있다. 그러나, 프라이버시가 고려사항일 때에는, SessionIndex 값이 다른 프라이버시 메커니즘을 무효화하지 않도록 주의를 기울여야만 한다. 따라서, 이 값은 다른 세션 참여자들(participants)에 걸친 사용자(principal) 행동을 상호연관(correlate) 시키는데 사용되지 않아야 한다. 이러한 목적을 달성하기 아래에 두 가지 솔루션(solutions)이 제공되며 그 사용이 권고된다.

- SessionIndex 의 값으로 작은 양의 정수들을 (또는 리스트 형태로 재발생하는 상수들) 사용한다. SAML 기관은 어떠한 하나의 정수의 카디널리티(cardinality)가 충분히 높아 한 사용자(principal)의 행동들이 다중 세션 참여자들에서 상호연관이 되는 것을 방지하도록 세션인덱스 값들의 범위를 선택해야 한다. (동일한 세션 참여자지만 다른 세션의 일부로써 주어지는 후속(subsequent) 문장들에 대하여 유일한 값들을 보장하는 것이 요구될 때를 제외하면) SAML 기관은 이런 범위 내에서 SessionIndex 에 대한 값을 임의로 선택해야 한다.
- 둘러싼(enclosing) 주장의 ID 를 SessionIndex 에 사용한다.

다음 스키마 조각은 <AuthnStatement> 요소와 **AuthnStatementType** 복합 타입을 정의한다.

```

<element name="AuthnStatement" type="saml:AuthnStatementType"/>

<complexType name="AuthnStatementType">
    <complexContent>
        <extension base="saml:StatementAbstractType">
            <sequence>
                <element ref="saml:SubjectLocality" minOccurs="0"/>
                <element ref="saml:AuthnContext"/>
            </sequence>
            <attribute name="AuthnInstant" type="dateTime" use="required"/>
            <attribute name="SessionIndex" type="string" use="optional"/>
            <attribute name="SessionNotOnOrAfter" type="dateTime"
                use="optional"/>
        </extension>
    </complexContent>
</complexType>

```

2.7.2.1. <SubjectLocality> 요소

<SubjectLocality> 요소는 주장 주체가 어느 시스템으로부터 인증이 되었는지를 나타내는 DNS 도메인 이름과 IP 주소 값을 기술한다. 이것은 다음 속성들을 가진다.

Address [Optional]

주체에 의해 식별되는 사용자(principal)가 어느 시스템으로부터 인증되었는지 나타내기 위해 그 시스템에 대한 네트워크 주소를 나타냄. IPv4 주소들은 점-10 진(dotted-decimal) 형식(예, "1.2.3.4")으로 표현되어야 한다. IPv6 주소들은 IETF RFC 3513 의 2.2 절에서 정의된 것처럼 (예, "FEDC:BA98:7654:3210:FEDC:BA98:7654:3210") 표현되어야 한다.

DNSName [Optional]

주체에 의해 식별되는 사용자(principal)가 어느 시스템으로부터 인증되었는지 나타내기 위해 시스템에 대한 DNS 이름을 나타냄.

이 두 필드 모두가 아주 쉽게 속여질 수(spoofed) 있기 때문에 이 요소는 전체적으로 충고적(advisory)이지만 일부 어플리케이션에서는 매우 유용한 정보가 될 수 있다.

다음 스키마 조각은 <SubjectLocality> 요소와 **SubjectLocalityType** 복합 타입을 정의한다.

```

<element name="SubjectLocality" type="saml:SubjectLocalityType"/>

<complexType name="SubjectLocalityType">
  <attribute name="Address" type="string" use="optional"/>
  <attribute name="DNSName" type="string" use="optional"/>
</complexType>

```

2.7.2.2. <AuthnContext> 요소

<AuthnContext> 요소는 인증 사건의 문맥을 기술한다. 이 요소는 인증 문맥 클래스 참조를 포함하거나 인증 문맥 선언 또는 선언 참조를 포함하거나 또는 둘 다를 포함한다. 이것의 타입은 AuthnContextType 이며 다음 요소들을 가진다.

<AuthnContextClassRef> [Optional]

인증기관이 따르는 인증 문맥 선언을 설명하는 인증 문맥 클래스를 식별하는 URI 참조.

<AuthnContextDecl> 또는 <AuthnContextDeclRef> [Optional]

값으로 제공되는 인증 문맥 선언 또는 그와 같은 선언을 식별하는 URI 참조. URI 참조는 직접적으로 참조된 선언을 포함하는 XML 문서로 해결(resolve) 될 수 있다.

<AuthenticatingAuthority> [Zero or More]

(명시적으로 명명되지 않으면서 인증업무와 관련되어 있다고 가정되는 주장 발급자는 포함하지 않으면서) 사용자(principal) 인증에 참여한 영 또는 그 이상의 인증 기관들의 유일한 식별자들

인증 문맥 정보에 대한 자세한 설명은 인증 문맥 표준을 참조한다.

다음 스키마 조각은 <SubjectLocality> 요소와 **SubjectLocalityType** 복합 타입을 정의한다.

```

<element name="AuthnContext" type="saml:AuthnContextType"/>

<complexType name="AuthnContextType">
  <sequence>
    <choice>
      <sequence>
        <element ref="saml:AuthnContextClassRef"/>
        <choice minOccurs="0">
          <element ref="saml:AuthnContextDecl"/>

```



```

<element ref="saml:AuthnContextDeclRef"/>
    </choice>
</sequence>
<choice>
    <element ref="saml:AuthnContextDecl"/>
    <element ref="saml:AuthnContextDeclRef"/>
    </choice>
</choice>
<element ref="saml:AuthenticatingAuthority" minOccurs="0"
    maxOccurs="unbounded"/>
</sequence>
</complexType>

<element name="AuthnContextClassRef" type="anyURI"/>
<element name="AuthnContextDeclRef" type="anyURI"/>
<element name="AuthnContextDecl" type="anyType"/>
<element name="AuthenticatingAuthority" type="anyURI"/>

```

2.7.3. <AttributeStatement> 요소

<AttributeStatement> 요소는 주장 주체가 명기된 속성들과 연관되어 있다는 것을 보장하는 SAML 기관에 의해 생성되는 문장을 설명한다. <AttributeStatement> 요소들을 포함하는 주장들은 <Subject> 요소를 반드시 포함해야만 한다.

이것은 **AttributeStatementType** 타입이며, 이 타입은 다음 요소들과 속성들을 추가하여 **StatementAbstractType** 을 확장한다:

<Attribute> 또는 <EncryptedAttributes> [One or More]

<Attribute> 요소는 주장 주체의 속성을 기술한다. 암호화된 SAML 속성은 <EncryptedAttribute> 요소에 포함될 수 있다.

다음 스키마 조각은 <AttributeStatement> 요소와 **AttributeStatementType** 복합 타입을 정의한다.

```

<element name="AttributeStatement" type="saml:AttributeStatementType"/>

<complexType name="AttributeStatementType">
    <complexContent>
        <extension base="saml:StatementAbstractType">
            <choice maxOccurs="unbounded">
                <element ref="saml:Attribute"/>
                <element ref="saml:EncryptedAttribute"/>
            </choice>
        </extension>
    </complexContent>
</complexType>

```

2.7.3.1. <Attribute> 요소

<Attribute> 요소는 이름으로 속성을 식별하며 선택적으로 그것의 값(들)을 포함한다. 이것은 **AttributeType** 복합 타입이다. 이것은 앞 절에서 설명한 것과 같이 주장 주체와 관련된 특정한 속성들과 값들을 표현하기 위해 속성 문장에서 사용된다. 이것은 특정한 SAML 속성들의 값들이 반환되도록 요청하는 속성 질의에서 또한 사용될 수 있다 (자세한 정보는 3.3.2.3 절을 참조). <Attribute> 요소는 다음 XML 속성들을 포함한다.

Name [Required]

속성의 이름

NameFormat [Required]

이름을 해석하기 위해 속성 이름에 대한 분류(classification)을 나타내는 URI 참조. NameFormat 속성의 값으로 사용될 수 있는 일부 URI 참조들과 그들과 관련된 설명 및 처리 규칙에 대한 내용은 8.2 절을 참조한다. 만약 NameFormat 값이 제공되지 않는다면, 식별자 urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified(8.2.1 절 참조)이 효력을 가진다.

FriendlyName [Optional]

인간이 읽기 편한 형태의 속성 이름을 제공하는 문자열. 이것은 실제 Name 이 OID 또는 UUID 와 같이 복잡하고 불투명한(opaque) 경우에 유용할 수 있다. 이 속성의 값은 형식적으로(formally) SAML 속성들을 식별하기 위한 기초로 사용되지 않아야만 한다.

임의의 속성들 (Arbitrary attributes)

이 복합 타입은 <xs:anyAttribute> 확장점을 사용하여, 명시적으로 스키마를 확장할 필요 없이, 임의의 XML 속성들이 <Attribute> 구조들에 추가되는 것을 허용한다. 이것은 예를 들어 속성 질의에서 추가적인 파라미터들이 사용되도록 제공하는 것이 필요할 때마다, 추가적인 필드들이 추가되는 것을 허용한다. SAML 확장들은 지역(local) (non-namespace-qualified) XML 속성들 또는 SAML 에서 정의된 네임스페이스로 제한되는 XML 속성들이 **AttributeType** 복합 타입 또는 그것으로 유도된 타입에 추가되지 않아야만 한다. 이와 같은 속성들은 SAML 자체에 대한 미래의 유지보수와 개선을 위해 예약되어 있다.

<AttributeValue> [Any Number]

속성의 값을 포함한다. 만약 속성이 하나 이상의 이산 값(discrete value)을 포함하면 각각의 값이 자신의 <AttributeValue> 요소에 나타나는 것이 권고된다. 만약 하나의 속성에 대하여 하나 이상의 <AttributeValue> 요소들이 제공되고 요소들 중에 하나라도 xsi:type 을 통해 할당된 데이터타입을 가지고 있다면, 모든 <AttributeValue> 요소들은 동일한 데이터타입이 할당되어야만 한다.

<AttributeValue> 요소를 포함하지 않는 <Attribute> 요소의 의미는 그것의 문맥에 따라 달리 해석된다. <AttributeStatement> 내에서는, 만약 SAML 속성이 존재하지만 어떠한 값도 갖고 있지 않으면, <AttributeValue> 요소는 반드시 생략되어야만 한다. <samlp:AttributeQuery> 내에서는, 값들이 존재하지 않으면, 이것은 요청자가 명명된 속성의 값들 중에 하나 또는 전부에 관심이 있다는 것을 가리킨다 (3.3.2.3 절을 참조).

프로파일 또는 다른 표준들에 의해 <Attribute> 요소가 다른 방식으로 사용되면, <AttributeValue> 요소들을 명기하거나 또는 생략하는 것에 대한 의미 또한 정의되어야만 한다.

다음 스키마 조각은 <Attribute> 요소와 **AttributeType** 복합 타입을 정의한다.

```
<element name="Attribute" type="saml:AttributeType"/>
```

```
<complexType name="AttributeType">
```

```
<sequence>
```

```
<element ref="saml:AttributeValue" minOccurs="0"
maxOccurs="unbounded"/>
```

```
</sequence>
```

```
<attribute name="Name" type="string" use="required"/>
```

```
<attribute name="NameFormat" type="anyURI" use="optional"/>
```

```
<attribute name="FriendlyName" type="string" use="optional"/>
```

```
<anyAttribute namespace="##other" processContents="lax"/>
```

```
</complexType>
```

<AttributeValue> 요소는 명기된 SAML 속성의 값을 제공한다. 이것은 xs:anyType 타입을 가지며, 이렇게 함으로써 어떠한 적격(well-formed) XML 도 이 요소의 내용으로 나타날 수 있도록 해 준다.

만약 <AttributeValue> 요소의 데이터 내용이 (xs:integer 또는 xs:string 과 같은) XML 스키마 단순 타입이면, 데이터타입이 <AttributeValue> 요소에서 xsi:type 선언을 통해 명확히 선언될 수 있다. 만약 속성 값이 구조화된 데이터(structured data)를 포함하면, 필요한 데이터 요소들은 확장 스키마에서 정의될 수 있다.

주의: xsi:type 을 이용하여 <AttributeValue>에 XML 스키마 단순 타입이 아닌 다른 데이터 타입을 명기하는 것은 스키마 처리의 진행을 위하여 해당 데이터타입을 정의하는 확장 스키마의 존재를 요구할 것이다.

만약 SAML 속성이 빈 문자열(empty string)과 같은 빈 값(empty value)을 포함하면, 대응되는 <AttributeValue> 요소는 반드시 비어(empty) 있어야만 한다 (일반적으로 이것은 <AttributeValue/> 로 직렬화(serialized) 된다).

만약 SAML 속성이 “null” 값을 포함하면, 대응되는 <AttributeValue> 요소는 반드시 비어(empty) 있어야만 하고 그 값으로 “true” 또는 “1”을 가지는 예약어 xsi:nil XML 속성을 포함해야만 한다.

다음 스키마 조각은 <AttributeValue> 요소를 정의한다.

```
<element name="AttributeValue" type="anyType" nillable="true"/>
```

2.7.3.2. <EncryptedAttribute> 요소

<EncryptedAttribute> 요소는 W3C XML Encryption 에서 정의된 것처럼 SAML 속성을 암호화된 방식으로 나타낸다. <EncryptedAttribute> 요소는 다음 요소들을 포함한다.

<xenc:EncryptedData> [Required]

W3C XML Encryption 에서 정의된 것처럼, 암호화된 내용과 관계된 암호화 세부사항. Type 속성은 존재해야 하며 만약 존재하면 반드시 <http://www.w3.org/2001/04/xmlenc#Element> 를 값으로 가져야만 한다. 암호화된 내용은 AttributeType 타입 또는 이것으로부터 유도된 타입을 가지는 요소를 포함해야만 한다.

<xenc:EncryptedKey> [Zero or More]

W3C XML Encryption 에서 정의된 것처럼, 포장된 복호화 키들(wrapped decryption keys). 각각의 포장된 키는 그것이 어떤 엔티티를 대상으로 암호화 되었는지를 나타내도록 엔티티를 기술하는 Recipient 속성을 포함해야 한다. Recipient 속성은 8.3.6 절에서 정의된 것처럼, SAML 시스템 엔티티에 대한 URI 식별자이어야 한다.

평문 값(plain-text value)이 중개자를 통해 전달될 때, 프라이버시 보호 메커니즘으로 암호화된 주장들이 이용된다.

다음 스키마 조각은 <EncryptedAttribute> 요소를 정의한다.

```
<element name="EncryptedAttribute" type="saml:EncryptedElementType"/>
```

2.7.4. <AuthzDecisionStatement> 요소

<AuthzDecisionStatement> 요소는 명기된 자원에 대한 주장 주체의 접근 요청이 일부 선택적으로 명기된 증거(specified evidence)를 근거로 하여 명기된 인가 결정을 발생시켰다는 것을 보장하는 SAML 기관에 의해 생성된 문장을 설명한다. <AuthzDecisionStatement> 요소를 포함하는 주장들은 반드시 <Subject> 요소를 포함해야만 한다.

자원은 URI 참조로 식별된다. 주장을 올바르게 안전하게 해석하기 위하여, SAML 기관과 SAML 을 의지하는 기관은 각각의 URI 참조를 일관된 방식으로 해석해야만 한다. 일관된 URI 참조 해석을 달성하는 것이 실패하면, 자원 URI 참조의 인코딩에 의존하는 인가 결정이 서로 다르게 내려질 수 있다. URI 참조들을 정규화하는 규칙들은 IETF RFC 2396 6 장에서 발견할 수 있다.

URI 인코딩 변이들에서 발생하는 모호성(ambiguity)를 회피하기 위해, SAML 시스템 엔티티들은 가능한 곳에서는 다음과 같이 URI 정규화된 형식 (URI normalized form)을 채택해야 한다

- SAML 기관들은 모든 자원 URI 참조들을 정규화된 형식으로 인코딩해야 한다
- 의지하는 기관들은 처리를 진행하기에 앞서 자원 URI 참조들을 정규화된 형식으로 변환해야 한다

URI 참조 문법과 하부 파일 시스템의 의미들 사이에서 차이가 발생하면 이것은 일관성없는(inconsistent) URI 참조 해석을 발생시킬 수 있다. 만약 URI 참조들이 접근 제어 정책 언어를 기술하기 위해 채택된다면 특별한 주의가 요구된다. 다음 보안 조건들은 SAML 주장들을 채택하는 시스템에서 충족되어야 한다.

- URI 참조 문법의 일부는 대소문자를 구분한다. 만약 하부의 파일 시스템이 대소문자를 구분하지 않으면, 요청자는 자원 URI 참조의 일부분에 대한 대소문자(case)를 변화함으로써 거절된 자원에 대한 접근을 얻을 수 있도록 하지 않아야 한다.
- 많은 파일 시스템들은 논리적 패스들과 심볼릭 링크들과 같은 메커니즘을 지원하여 사용자들이 파일 시스템 엔트리들 사이에서 논리적인 등치성들을 설정하는 것을 허용한다. 요청자가 이와 같은 등치성을 생성함으로써 거절된 자원에 대한 접근을 얻을 수 있도록 하지 않아야 한다.

<AuthzDecisionStatement> 요소는 **AuthzDecisionStatementType** 타입이며, 이 타입은 다음 요소들과 속성들을 추가하여 **StatementAbstractType** 을 확장한다:

Resource [Required]

접근 인가가 구해지는 자원을 식별하는 URI 참조. 이 속성은 빈 URI 참조 (")를 값으로 가질 수 있으며 그 의미는 IETF RFC 2396 4.2 절에 기술된 것과 같이, "현재 문서의 시작"으로 정의되어 있다.

Decision [Required]

명기된 자원에 대하여 SAML 기관에 의해 결정되는 판단. 이 값은 **DecisionType** 단순 타입을 가진다.

<Action> [One or more]

명기된 자원에 대하여 수행되도록 인가된 행동들의 집합

<Evidence> [Optional]

SAML 기관이 판단을 내릴 때 의지한 주장들의 집합

다음 스키마 조각은 <AuthzDecisionStatement> 요소와 **AuthzDecisionStatementType** 복합 타입을 정의한다.

```

<element name="AuthzDecisionStatement"
  type="saml:AuthzDecisionStatementType"/>

<complexType name="AuthzDecisionStatementType">
  <complexContent>
    <extension base="saml:StatementAbstractType">
      <sequence>
        <element ref="saml:Action" maxOccurs="unbounded"/>
        <element ref="saml:Evidence" minOccurs="0"/>
      </sequence>
      <attribute name="Resource" type="anyURI" use="required"/>
      <attribute name="Decision" type="saml:DecisionType" use="required"/>
    </extension>
  </complexContent>
</complexType>

```

2.7.4.1. DecisionType 단순 타입

DecisionType 단순 타입은 인가 결정문의 상태로 보고되는 것이 가능한 값들을 정의한다.

Permit

명기된 행동이 허용된다.

Deny

명기된 행동이 거절된다.

Indeterminate

SAML 기관은 명기 행동이 허용되는지 또는 거절되는지에 대하여 판단할 수 없다.

Indeterminate 결정 값은 SAML 기관이 긍정적인 문장을 제공할 능력을 요구하지만 어떠한 결정을 할 수는 없는 상황에서 사용된다. 결정을 제공하는 것을 거절하거나 또는 제공할 수 없는 이유에 대한 추가적인 정보는 포함하는 <Response>에서 <StatusDetail> 요소들로서 반환될 수 있다.

다음 스키마 조각은 **DecisionType** 단순 타입을 정의한다.

```
<simpleType name="DecisionType">
  <restriction base="string">
    <enumeration value="Permit"/>
    <enumeration value="Deny"/>
    <enumeration value="Indeterminate"/>
  </restriction>
</simpleType>
```

2.7.4.2. <Action> 요소

<Action> 요소는 허가가 구해지는 명기된 자원에 대해 행해질 행동에 대하여 명기한다. 이것의 문자열-데이터 내용은 명기된 자원에서 수행되기를 원하는 행동에 대한 라벨(label)을 제공한다. 이것은 다음 속성들을 가진다.

Namespace [Optional]

명기된 행동의 이름이 해석되는 네임스페이스를 나타내는 URI 참조. 만약 이 요소가 존재하지 않으면, 8.1.2 절에 기술된 urn:oasis:names:tc:SAML:1.0:action:rwdc-negation 네임스페이스가 효력을 발휘한다.

다음 스키마 조각은 <Action> 요소와 **ActionType** 복합 타입을 정의한다.

```

<element name="Action" type="saml:ActionType"/>

<complexType name="ActionType">
    <simpleContent>
        <extension base="string">
            <attribute name="Namespace" type="anyURI" use="required"/>
        </extension>
    </simpleContent>
</complexType>

```

2.7.4.3. <Evidence> 요소

<Evidence> 요소는 인가 결정을 발급할 때 SAML 기관이 의지하는 하나 또는 그 이상의 주장 또는 주장 참조들을 포함한다. 이것은 **EvidenceType** 복합 타입이다. 이것은 다음 요소들의 하나 또는 그 이상의 혼합을 포함한다.

<AssertionIDRef> [Any number]

주장의 ID 속성의 값에 대한 참조를 통해 주장을 명기한다.

<AssertionURIRef> [Any number]

URI 참조를 이용하여 주장을 명기한다.

<Assertion> [Any number]

주장을 값으로 명기한다.

<EncryptedAssertion> [Any number]

암호화된 주장을 값으로 명기한다.

주장을 증거로 제공하는 것은 SAML 의지하는 기관과 인가 결정을 내리는 SAML 기관 사이의 신뢰 협정에 영향을 줄 수 있다. 예를 들어, SAML 의지하는 기관이 SAML 기관에게 요청을 할 때 주장을 제출하는 경우에, SAML 기관은 의지하는 기관 또는 어떠한 제 3의 기관에게 <Evidence> 요소의 주장이 유효하다는 것을 보장하지 않으면서, 그 주장을 자신이 인가 결정을 내릴 때의 증거로써 사용할 수 있다.

다음 스키마 조각은 <Evidence> 요소와 **EvidenceType** 복합 타입을 정의한다.


```
<element name="Evidence" type="saml:EvidenceType"/>
```

```
<complexType name="EvidenceType">
```

```
<choice maxOccurs="unbounded">
```

```
<element ref="saml:AssertionIDRef"/>
```

```
<element ref="saml:AssertionURIRef"/>
```

```
<element ref="saml:Assertion"/>
```

```
<element ref="saml:EncryptedAssertion"/>
```

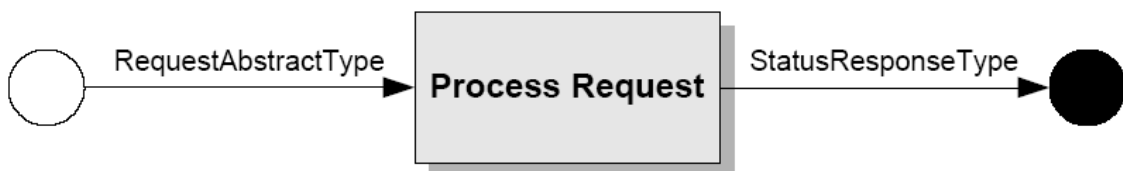
```
</choice>
```

```
</complexType>
```

3. SAML 프로토콜

SAML 프로토콜 메시지들은 다양한 프로토콜들을 사용하여 생성되고 교환될 수 있다. SAML 2.0 바인딩 표준은 기존에 널리 사용되고 있는 전송 프로토콜들을 이용하여 프로토콜 메시지를 전달하는(transporting) 특정한 수단들을 설명한다. SAML 2.0 프로파일 표준은 이 장에서 정의된 프로토콜들에 대한 많은 응용들(applications)을 설명하고 더불어 추가적인 규칙들, 제약들 그리고 상호운용을 쉽게 하기 위한 요구사항들에 대하여 설명한다.

특정한 SAML 요청과 응답 메시지들은 공통 타입들로부터 유도된다. 요청자는 **RequestAbstractType**로부터 유도된 요소를 SAML 응답자에게 전달하고, 응답자는 그림 1 에서 보여지는 것과 같이 **StatusResponseType** 타입을 가지거나 또는 그것으로부터 유도된 요소를 생성한다.



(그림 3-1) SAML 요청-응답 프로토콜

어떤 경우에는, 프로파일들에 의해 허용되면, 응답자가 대응되는 요청을 받지도 않고 SAML 응답을 생성하여 송신할 수 있다.

SAML 에서 정의되는 프로토콜은 다음 동작들을 달성한다:

- 하나 또는 그 이상의 요청된 주장들을 반환하기. 이것은 특정한 주장들에 대한 직접적인 요청이나 또는 특정한 조건을 충족하는 주장들에 대한 질의에 대한 응답으로 발생할 수 있다.
- 요청에 따라 인증을 수행하고 대응되는 주장을 반환하기
- 이름 식별자를 등록하거나 또는 요청에 따라 이름 등록을 종료하기
- 요청에 따라 관련된 세션 집합에 대하여 거의 동시적인 로그아웃을 수행하기 ("single logout")
- 요청에 따라 이름 식별자 매핑을 제공하기

이 장 전체에 걸쳐, SAML 프로토콜 네임스페이스에 속하는 요소들과 타입들에 대한 텍스트 설명들은 전통적인 네임스페이스 접두사인 `samlp:` 를 표시하지 않는다. 명확성을 위해, SAML 주장 네임스페이스에 속하는 요소들과 타입들의 텍스트 설명들은 전통적인 접두사인 `saml:` 로 표시된다.

3.1. 스키마 헤더와 네임스페이스 선언 (Schema Header and Namespace Declarations)

다음 스키마 조각은 프로토콜 스키마에 대한 XML 네임스페이스들과 다른 헤더 정보를 정의한다.

```
<schema
  targetNamespace="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  elementFormDefault="unqualified"
  attributeFormDefault="unqualified"
  blockDefault="substitution"
  version="2.0">

  <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
    schemaLocation="saml-schema-assertion-2.0.xsd"/>
  <import namespace="http://www.w3.org/2000/09/xmldsig#"
    schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
      20020212/xmldsig-core-schema.xsd"/>

  <annotation>
  <documentation>
```

Document identifier: saml-schema-protocol-2.0
 Location: <http://docs.oasis-open.org/security/saml/v2.0/>
 Revision history:
 V1.0 (November, 2002):
 Initial Standard Schema.
 V1.1 (September, 2003):
 Updates within the same V1.0 namespace.
 V2.0 (March, 2005):
 New protocol schema based in a SAML V2.0 namespace.
 </documentation>
 </annotation>
 ...
 </schema>

3.2. 요청들과 응답들 (Requests and Responses)

이번 절은 SAML 프로토콜들에서 사용되는 모든 요청과 응답 메시지들의 기초가 되는 SAML 구조들과 기본 요구사항들을 정의한다.

3.2.1. RequestAbstractType 복합 타입

모든 SAML 요청들은 추상적인 RequestAbstractType 복합 타입으로부터 유도된 타입을 가진다. 이 타입은 모든 SAML 요청들과 관련된 공통적인 속성들과 요소들을 정의한다.

주의: <ResponseWith> 요소는 SAML 2.0 에서 RequestAbstractType 으로부터 제거되었다.

ID [Required]

요청의 식별자. 이 요소의 타입은 **xs:ID** 이고 식별자 유일성을 위해 반드시 1.6.4 절에 명기된 요구조건을 따라야만 한다. 요청에 있는 ID 속성 값과 그 요청에 대응되는 응답의 InResponseTo 속성의 값들은 서로 일치(match)되어야만 한다.

Version [Required]

요청의 버전. 이 표준에서 정의된 SAML 의 버전을 위한 식별자는 “2.0” 이다.

IssueInstant [Required]

요청의 발급 시각. 시각 값은 1.6.3 절에서 설명된 것처럼 UTC 형식으로 인코딩된다.

Destination [Optional]

이 요청이 전달되는 목적지 주소를 가리키는 URI 참조. 이것은 의도되지 않은 수신자들에게 악의적으로 요청들을 전달하는 것을 막는데 유용하며, 일부 프로토콜 바인딩들에서 요구되는 보호 방법이다. 만약 이 속성이 존재하면, 실질적인 수신자는 URI 참조가 메시지가 수신된 위치를 식별하는지를 검사해야만 한다. 만약 그렇지 않다면, 요청은 폐기되어야만 한다. 일부 프로토콜 바인딩은 이 속성의 사용을 요구할 수 있다(SAML 2.0 바인딩 표준 참조).

Consent [Optional]

(어떠한 조건 하에) 이 요청을 송신할 때 사용자로부터 동의를 얻었는지 또는 얻지 않았는지를 가리킨다. Consent 속성의 값으로 사용될 수 있는 일부 URI 참조들과 그들과 연관된 설명들은 8.4 절을 참조한다. 만약 Consent 값이 제공되지 않는다면, 식별자 urn:oasis:names:tc:SAML:2.0:consent:unspecified 가 효력을 갖는다.

<saml:Issuer> [Optional]

요청 메시지를 생성한 엔티티를 식별한다.

<ds:Signature> [Optional]

아래 부분과 5 장에서 설명하는 것과 같이, 요청자를 인증하고 메시지 무결성을 제공하는 XML 서명

<Extensions> [Optional]

이 확장점은 통신 기관들 사이에 동의된 선택적인 프로토콜 메시지 확장 요소들을 포함한다. 이 확장점을 사용하기 위하여 어떠한 확장 스키마도 필요하지 않으며, 비록 확장 스키마가 제공되어도, 느슨한 검증 설정(lax validation setting)은 확장 내용이 유효할 것을 요구하지 않는다. SAML 확장 요소들은 SAML 에서 정의되지 않은 네임스페이스로 네임스페이스-한정되어야 한다.

특정한 프로토콜들과 프로파일들의 요구사항에 따라, SAML 요청자는 종종 그 자신을 인증시킬 필요가 있을 수 있고 메시지 무결성이 종종 요구될 수 있다. 인증과 메시지 무결성은 프로토콜 바인딩에서 제공되는 메커니즘들에 의해 제공될 수 있다(SAML 2.0 바인딩 표준 참조). SAML 요청은 서명될 수 있으며, 이것은 요청자에 대한 인증과 메시지 무결성을 동시에 제공한다.

만약 이와 같이 서명이 사용되면, <ds:Signature> 요소는 반드시 존재해야만 하고 SAML 응답자는 반드시 서명이 W3C XML Signature 에 따라 유효하다는 것(즉, 메시지가 변조(tamper) 되지 않았다)을 확인해야만 한다. 만약 이것이 유효하지 않으면, 응답자는 요청의 내용을 믿지 않아야만 하고 에러를 가지고 응답해야 한다. 만약 이것이 유효하면,

응답자는 서명자의 신원과 적절성을 판단하기 위해 서명을 평가해야 하고 요청을 계속적으로 처리하거나 또는 (만약 요청이 어떤 다른 이유로 유효하지 않다면) 에러를 가지고 응답할 수 있다.

만약, Consent 속성이 포함되고 그 값이 어떤 형태의 사용자 동의가 얻어졌다는 것을 가리킨다면, 요청은 서명되어야 한다.

만약 SAML 응답자가 SAML 문법 또는 처리 규칙에 따라 요청자가 유효하지 않다고 생각한다면, 그리고 만약 응답자가 응답한다면, 응답자는 urn:oasis:names:tc:SAML:2.0:status:Requester 값을 가지는 <StatusCode> 요소를 가지는 SAML 응답 메시지를 반환해야만 한다. 예를 들어 DoS 공격이 의심되는 동안과 같은, 일부 경우에는 응답하는 것이 전혀 보장되지 않을 지도 모른다.

다음 스키마 조각은 **RequestAbstractType** 복합 타입을 정의한다.

```
<complexType name="RequestAbstractType" abstract="true">
  <sequence>
    <element ref="saml:Issuer" minOccurs="0"/>
    <element ref="ds:Signature" minOccurs="0"/>
    <element ref="samlp:Extensions" minOccurs="0"/>
  </sequence>
  <attribute name="ID" type="ID" use="required"/>
  <attribute name="Version" type="string" use="required"/>
  <attribute name="IssueInstant" type="dateTime" use="required"/>
  <attribute name="Destination" type="anyURI" use="optional"/>
  <attribute name="Consent" type="anyURI" use="optional"/>
</complexType>

<element name="Extensions" type="samlp:ExtensionsType"/>

<complexType name="ExtensionsType">
  <sequence>
    <any namespace="##other" processContents="lax"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

3.2.2. StatusResponseType 복합 타입

모든 SAML 응답들은 추상적인 **StatusResponseType** 복합 타입으로부터 유도된 타입을 가진다. 이 타입은 모든 SAML 응답들과 관련된 공통적인 속성들과 요소들을 정의한다:

ID [Required]

응답에 대한 식별자. 이것은 **xs:ID** 타입이고, 식별자 유일성을 위해 반드시 1.6.4 절에 명기된 요구조건을 따라야만 한다.

InResponseTo [Optional]

만약에 존재한다면, 이 응답과 대응되는 요청의 식별자에 대한 참조. 만약 응답이 요청에 대한 응답으로 생성되지 않는다면 또는 요청의 ID 속성 값이 결정될 수 없다면 (예를 들어 요청이 비정상적인 형태(malformed)이다), 그러면 이 속성은 존재하지 않아야만 한다. 만약 이렇지 않다면, 이 속성은 반드시 존재해야만 하고 그것의 값은 대응되는 요청의 ID 속성의 값과 일치해야만 한다.

Version [Required]

응답의 버전. 이 표준에서 정의된 SAML의 버전을 위한 식별자는 “2.0”이다.

IssueInstant [Required]

응답의 발급 시각. 시각 값은 1.6.3 절에서 설명된 것처럼 UTC 형식으로 인코딩된다.

Destination [Optional]

이 응답이 전달되는 목적지 주소를 가리키는 URI 참조. 이것은 의도되지 않은 수신자들에게 악의적으로 응답들을 전달하는 것을 막는데 유용하며, 일부 프로토콜 바인딩들에서 요구되는 보호 방법이다. 만약 이 속성이 존재하면, 실질적인 수신자는 URI 참조가 메시지가 수신된 위치를 식별하는지를 검사해야만 한다. 만약 그렇지 않다면, 요청은 폐기되어야만 한다. 일부 프로토콜 바인딩은 이 속성의 사용을 요구할 수 있다(SAML 2.0 바인딩 표준 참조).

Consent [Optional]

(어떤 조건 하에) 이 응답을 송신할 때 사용자로부터 동의를 얻었는지 또는 얻지 못했는지를 가리킨다. Consent 속성의 값으로 사용될 수 있는 일부 URI 참조들과 그들과 연관된 설명들은 8.4 절을 참조한다. 만약 Consent 값이 제공되지 않는다면, 식별자 `urn:oasis:names:tc:SAML:2.0:consent:unspecified` 가 효력을 갖는다.

<saml:Issuer> [Optional]

응답 메시지를 생성한 엔티티를 식별한다. (이 요소에 대한 더 많은 정보를 얻으려면, 2.2.5 절을 참조)

<ds:Signature> [Optional]

아래 부분과 5 장에서 설명하는 것과 같이, 응답자를 인증하고 메시지 무결성을 제공하는 XML 서명

<Extensions> [Optional]

이 확장점은 통신 기관들 사이에 동의된 선택적인 프로토콜 메시지 확장 요소들을 포함한다. 이 확장점을 사용하기 위해서 어떠한 확장 스키마도 필요하지 않으며, 비록 확장 스키마가 제공되어도, 느슨한 검증 설정(lax validation setting)은 확장 내용이 유효할 것을 요구하지 않는다. SAML 확장 요소들은 SAML 에서 정의되지 않은 네임스페이스로 네임스페이스-한정되어야 한다.

<Status> [Required]

요청의 처리 상태를 표현하는 코드

특정한 프로토콜들과 프로파일들의 요구사항에 따라, SAML 응답자는 종종 그 자신을 인증시킬 필요가 있을 수 있고 메시지 무결성이 종종 요구될 수 있다. 인증과 메시지 무결성은 프로토콜 바인딩에서 제공되는 메커니즘들에 의해 제공될 수 있다. SAML 응답은 서명될 수 있으며, 이것은 응답자에 대한 인증과 메시지 무결성을 동시에 제공한다.

만약 이와 같이 서명이 사용되면, <ds:Signature> 요소는 반드시 존재해야만 하고 응답을 수신하는 요청자는 반드시 서명이 W3C XML Signature 에 따라 유효하다는 것(즉, 메시지가 변조(tamper) 되지 않았다는)을 확인해야만 한다. 만약 이것이 유효하지 않으면, 요청자는 응답의 내용을 믿지 않아야만 하고 응답을 에러로 처리해야 한다. 만약 이것이 유효하면, 요청자는 서명자의 신원과 적절성을 판단하기 위해 서명을 평가해야 하고 그것이 적절하다고 생각될 때 응답의 처리를 계속할 수 있다.

만약, Consent 속성이 포함되고 그 값이 어떤 형태의 사용자 동의가 얻어졌다는 것을 가리킨다면, 응답은 서명되어야 한다.

다음 스키마 조각은 **StatusResponseType** 복합 타입을 정의한다.

```
<complexType name="StatusResponseType">
  <sequence>
    <element ref="saml:Issuer" minOccurs="0"/>
    <element ref="ds:Signature" minOccurs="0"/>
    <element ref="samlp:Extensions" minOccurs="0"/>
    <element ref="samlp:Status"/>
  </sequence>
  <attribute name="ID" type="ID" use="required"/>
  <attribute name="InResponseTo" type="NCName" use="optional"/>
</complexType>
```

```

<attribute name="Version" type="string" use="required"/>
<attribute name="IssueInstant" type="dateTime" use="required"/>
<attribute name="Destination" type="anyURI" use="optional"/>
<attribute name="Consent" type="anyURI" use="optional"/>
</complexType>

```

3.2.2.1. <Status> 요소

<Status> 요소는 다음과 같은 요소들을 포함한다:

<StatusCode> [Required]

대응되는 요청에 대한 응답으로 수행된 활동(activity)의 상태를 나타내는 코드

<StatusMessage> [Optional]

오퍼레이터(operator)에게 반환될 수 있는 메시지

<StatusDetail> [Optional]

상태에 대한 추가적인 정보

다음 스키마 조각은 **StatusResponseType** 복합 타입을 정의한다.

```

<element name="Status" type="sampl:StatusType"/>

<complexType name="StatusType">
  <sequence>
    <element ref="sampl:StatusCode"/>
    <element ref="sampl:StatusMessage" minOccurs="0"/>
    <element ref="sampl:StatusDetail" minOccurs="0"/>
  </sequence>
</complexType>

```

3.2.2.2. <StatusCode> 요소

<StatusCode> 요소는 대응되는 요청의 처리 상태를 나타내는 코드 또는 중첩된(nested) 코드들의 집합을 명기한다. <StatusCode> 요소는 다음 요소들을 포함한다:

Value [Required]

상태 코드 값. 이 속성은 URI 참조를 포함한다. 최상위(topmost) <StatusCode> 요소의 값은 반드시 이 절에서 제공되는 최상위 수준(top-level) 리스트에 존재하는 값이어야만 한다.

<StatusCode> [Optional]

에러 조건에 대한 더 자세한 정보를 제공하는 하위 상태 코드. 응답자는 고의적으로 잘못된 요청을 전달하여 부가적인 정보를 탐지하려는 공격을 방지하기 위해 하위 상태 코드들을 생략할 수 있다.

최상위 수준 <StatusCode> 값으로 허용되는 값들은 다음과 같다.

urn:oasis:names:tc:SAML:2.0:status:Success

요청이 성공했다. 추가적인 정보는 <StatusMessage> 그리고/또는 <StatusDetail> 요소들로 반환될 수 있다.

urn:oasis:names:tc:SAML:2.0:status:Requester

요청이 요청자 측의 에러 때문에 수행될 수 없었다.

urn:oasis:names:tc:SAML:2.0:status:Responder

요청이 SAML 응답자 또는 SAML 기관 측의 에러 때문에 수행될 수 없었다.

urn:oasis:names:tc:SAML:2.0:status:VersionMismatch

요청 메시지의 버전이 올바르지 않기 때문에 SAML 응답자가 요청을 처리할 수 없었다.

다음 차상위 수준(second-level) 상태 정보들은 이 표준의 다양한 곳에서 참조된다. 추가적인 차상위 수준 상태 코드들이 SAML 표준의 미래 버전들에서 정의될 수 있다. 시스템 엔티티들은 자유롭게 적절한 URI 참조들을 정의함으로써 더욱 더 한정적인(specific) 상태 코드들을 정의할 수 있다.

urn:oasis:names:tc:SAML:2.0:status:AuthnFailed

응답 제공자(responding provider)가 성공적으로 사용자를 인증할 수 없었다.

urn:oasis:names:tc:SAML:2.0:status:InvalidAttrNameOrValue

예기치 못한 또는 유효하지 못한 내용이 <saml:Attribute> 또는 <saml:AttributeValue> 요소 내에서 존재하였다.

urn:oasis:names:tc:SAML:2.0:status:InvalidNameIDPolicy

응답 제공자가 요청된 이름 식별자 정책을 지원할 수 없거나 또는 지원하지 않을 것이다.

urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext

명기된 인증 문맥 요구조건이 응답자에 의해 충족될 수 없다.

urn:oasis:names:tc:SAML:2.0:status:NoAvailableIDP

지원되는 IdP(Identity Provider)들 중 어느 것도 <IDPList> 내의 <Loc>로 해결될 수 없거나 또는 지원되는 IdP 들 중에 어느 것도 현재 이용 가능하지 않다는 것을 중개자가 가리키려 할 때 사용된다.

urn:oasis:names:tc:SAML:2.0:status:NoPassive

응답 제공자가 사용자를 수동적으로 인증할 수가 없다는 것을 가리킨다.

urn:oasis:names:tc:SAML:2.0:status:NoSupportedIDP

<IDPList>에 있는 IdP 들 중에 어떤 것도 중개자가 지원하지 않는다는 것을 가리키기기 위해 중개자에 의해 사용된다.

urn:oasis:names:tc:SAML:2.0:status:PartialLogout

세션 기관(session authority)이 세션 참여자(session participant)에게 로그아웃을 다른 모든 세션 참여자들에게 전파(propagate)시킬 수 없다는 것을 알릴 때 사용된다.

urn:oasis:names:tc:SAML:2.0:status:ProxyCountExceeded

응답 제공자가 사용자를 직접 인증할 수 없고 이 요청을 더 이상 대리하는(proxy) 것이 허용되지 않는다는 것을 가리킨다.

urn:oasis:names:tc:SAML:2.0:status:RequestDenied

SAML 응답자 또는 SAML 기관이 요청을 처리할 수는 있지만 응답하지 않기로 선택했다. 이 상태 코드는 특정한 요청자로부터 수신된 요청 메시지 또는 일련의 요청 메시지에 대한 보안 문맥에 대한 우려가 있을 때 사용될 수 있다.

urn:oasis:names:tc:SAML:2.0:status:RequestUnsupported

SAML 응답자 또는 SAML 기관은 요청을 지원하지 않는다.

urn:oasis:names:tc:SAML:2.0:status:RequestVersionDeprecated

SAML 응답자는 요청에서 명기된 프로토콜 버전을 가진 어떠한 요청들도 처리할 수 없다.

urn:oasis:names:tc:SAML:2.0:status:RequestVersionTooHigh

요청 메시지에 명기된 프로토콜 버전이 응답자에 의해 지원되는 가장 높은 프로토콜 버전보다도 더 높은 것이기 때문에 SAML 응답자는 요청을 처리할 수 없다.

urn:oasis:names:tc:SAML:2.0:status:RequestVersionTooLow

요청 메시지에 명기된 프로토콜 버전이 너무 낮아 SAML 응답자가 요청을 처리할 수 없다.

urn:oasis:names:tc:SAML:2.0:status:ResourceNotRecognized

요청 메시지에서 제공된 자원 값이 유효하지 않거나 또는 인식되지 않는 것이다.

urn:oasis:names:tc:SAML:2.0:status:TooManyResponses

응답 메시지가 SAML 응답자가 반환할 수 있는 것보다 더 많은 요소들을 포함하려 하였다.

urn:oasis:names:tc:SAML:2.0:status:UnknownAttrProfile

특정한 속성 프로파일에 대해 알지 못하는 엔티티에게 그 프로파일로부터 유도된 속성이 제출되었다.

urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal

응답 제공자가 요청에 의해 명기되거나 또는 암시된 사용자를 인식하지 못 한다.

urn:oasis:names:tc:SAML:2.0:status:UnsupportedBinding

SAML 응답자가 요청에 명기된 프로토콜 바인딩을 사용하여 요청의 처리를 적절하게 완료할 수 없다.

다음 스키마 조각은 <StatusCode> 요소와 **StatusCodeType** 복합 타입을 정의한다.

```
<element name="StatusCode" type="samlp:StatusCodeType"/>
```

```
<complexType name="StatusCodeType">
```

```
<sequence>
```

```
<element ref="samlp:StatusCode" minOccurs="0"/>
```

```
</sequence>
```

```
<attribute name="Value" type="anyURI" use="required"/>
```

```
</complexType>
```

3.2.2.3. <StatusMessage> 요소

<StatusMessage> 요소는 오퍼레이터에게 반환될 수 있는 메시지를 명기한다.

다음 스키마 조각은 <StatusMessage> 요소를 정의한다.

```
<element name="StatusMessage" type="string"/>
```

3.2.2.4. <StatusDetail> 요소

<StatusDetail> 요소는 요청의 처리 상태에 대해 추가적인 정보를 명기하기 위해 사용될 수 있다. 추가적인 정보는 영 또는 그 이상의 요소로 구성되며 구성 요소들은 any 네임스페이스를 가진다. 스키마가 제공되어야 한다거나 <StatusDetail> 내용들의 스키마 검증(validation)이 필요하다는 요구조건은 없다.

다음 스키마 조각은 <StatusDetail> 요소와 **StatusDetailType** 복합 타입을 정의한다.

```

<element name="StatusDetail" type="samlp:StatusDetailType"/>

<complexType name="StatusDetailType">
    <sequence>
        <any namespace="##any" processContents="lax" minOccurs="0"
            maxOccurs="unbounded"/>
    </sequence>
</complexType>

```

3.3. 주장 질의와 요청 프로토콜 (Assertion Query and Request Protocol)

이번 절은 참조에 의해 이미 존재하는 주장들을 요청하거나 또는 주체와 문장 타입으로 주장들을 질의하는데 필요한 메시지들과 처리 규칙들을 정의한다.

3.3.1. <AssertionIDRequest> 요소

만약 요청자가 하나 또는 그 이상의 주장들에 대한 유일한 식별자를 안다면, 주장들이 <Response> 메시지를 통해 반환되도록 요청하는데 <AssertionIDRequest> 메시지 요소가 사용될 수 있다. <saml:AssertionIDRef> 요소는 반환될 각각의 주장을 기술하는데 사용된다. 이 요소에 대한 더 많은 정보를 위해서 2.3.1 절을 참조한다.

다음 스키마 조각은 <AssertionIDRequest>를 정의한다.

```

<element name="AssertionIDRequest" type="samlp:AssertionIDRequestType"/>

<complexType name="AssertionIDRequestType">
    <complexContent>
        <extension base="samlp:RequestAbstractType">
            <sequence>
                <element ref="saml:AssertionIDRef" maxOccurs="unbounded"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

3.3.2. 질의들 (Queries)

이번 절은 SAML 질의 요청 메시지들을 정의한다.

3.3.2.1. <SubjectQuery> 요소

<SubjectQuery> 요소는 하나의 단일 SAML 주체를 명기하는 새로운 SAML 질의들이 정의되는 것을 허용하는 확장점이다. 이것의 **SubjectQueryAbstractType** 복합 타입은 추상적이고 따라서 유도 타입의 기본(base)으로만 사용될 수 있다. **SubjectQueryAbstractType** 은 (2.4 절에 정의된) <saml:Subject> 요소를 **RequestAbstractType** 에 추가한다.

다음 스키마 조각은 <SubjectQuery> 요소와 **SubjectQueryAbstractType** 복합 타입을 정의한다.

```
<element name="SubjectQuery" type="samlp:SubjectQueryAbstractType"/>

<complexType name="SubjectQueryAbstractType" abstract="true">
  <complexContent>
    <extension base="samlp:RequestAbstractType">
      <sequence>
        <element ref="saml:Subject"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

3.3.2.2. <AuthnQuery> 요소

<AuthnQuery> 메시지 요소는 “인증 문장을 포함하는 어떠한 주장들이 이 주체에 대하여 이용가능한가?” 질의를 만들 때 사용된다. 성공적인 <Response>는 인증 문장들을 포함하는 하나 또는 그 이상의 주장들을 포함할 것이다.

<AuthnQuery> 메시지는 요청에서 제공된 크리덴셜들(credentials)을 사용하여 새로운 인증을 요청하는데 사용되지 않아야만 한다. <AuthnQuery>는 가리켜지는 주체와 인증 기관 사이에서 이전의 상호작용에서 발생한 인증 동작들(authentication acts)에 대한 문장들을 요청하는 것이다.

이 요소는 **AuthnQueryType** 타입이며, 이것은 다음 요소와 속성을 추가하여 **SubjectQueryAbstractType** 을 확장한다.

SessionIndex [Optional]

만약 존재하면, 가능한 응답들에 대한 필터(filter)를 명기한다. 이와 같은 질의는 “제공된 세션 정보 문맥 내에서 이 주체에 대한 인증 문장들을 포함하는 어떤 주장들을 당신은 가지고 있습니까?”와 같은 질문을 물어본다.

<RequestedAuthnContext> [Optional]

만약 존재하면, 가능한 응답들에 대한 필터(filter)를 명기한다. 이와 같은 질의는 “이 주체에 대하여 이 요소 내의 인증 문맥 요구사항들을 만족시키는 인증 문장들을 포함하는 어떤 주장들을 당신은 가지고 있습니까?”와 같은 질문을 물어본다.

인증 질의에 대한 응답으로, SAML 기관은 다음과 같이 인증 문장들을 가지는 주장들을 반환한다.

- 질의의 <Subject> 요소와 매칭을 위하여 3.3.4 절에 주어진 규칙들은 반환될 수 있는 주장들을 식별한다.
 - 만약, SessionIndex 속성이 질의에 존재하면, 반환되는 주장 집합에서 적어도 하나의 <AuthnStatement> 요소는 반드시 질의의 SessionIndex 속성과 일치하는 SessionIndex 속성을 포함해야만 한다. 이와 같이 일치(match)하는 모든 주장들을 응답으로 반환하도록 하는 것은 선택적이다.
 - 만약 <RequestedAuthnContext> 요소가 질의에 존재하면, 반환되는 주장 집합에서 적어도 하나의 <AuthnStatement> 요소는 반드시 질의의 <RequestAuthnContext>를 충족시키는 <AuthnContext> 요소를 포함해야만 한다 (3.3.2.2.1 절을 참조). 이와 같이 일치(match)하는 모든 주장들이 응답으로 반환하도록 하는 것은 선택적이다.
- 다음 스키마 조각은 <AuthnQuery> 요소와 **AuthnQueryType** 복합 타입을 정의한다.

```
<element name="AuthnQuery" type="samlp:AuthnQueryType"/>

<complexType name="AuthnQueryType">
  <complexContent>
    <extension base="samlp:SubjectQueryAbstractType">
      <sequence>
        <element ref="samlp:RequestedAuthnContext" minOccurs="0"/>
      </sequence>
      <attribute name="SessionIndex" type="string" use="optional"/>
    </extension>
  </complexContent>
</complexType>
```

3.3.2.2.1. <RequestedAuthnContext> 요소

<RequestedAuthnContext> 요소는 요청 또는 질의에 대한 응답으로 반환되는 인증 문장들의 인증 문맥 요구사항들을 기술한다. 이것의 **RequestedAuthnContextType** 복합타입은 다음 요소들과 속성들을 정의한다.

<saml:AuthnContextClassRef> 또는 <saml:AuthnContextDeclRef> [One or More]

인증 문맥 클래스들 또는 선언들을 식별하는 하나 또는 그 이상의 URI 참조들을 명기한다. 이 요소들은 2.7.2.2 절에 정의되어 있다. 인증 문맥 클래스들에 대한 더 많은 정보는 SAML 인증 문맥 표준을 참조한다.

Comparison [Optional]

요청되는 문맥 클래스들 또는 문장들을 평가하는데 사용되는 비교 방법을 기술한다. "exact", "minimum", "maximum", 또는 "better" 중에 하나의 값을 가지며 기본(default)은 "exact" 이다.

클래스 참조들의 집합 또는 선언 참조들의 집합 둘 중에 하나가 사용될 수 있다. 제공되는 참조들의 집합은 순서를 가진 집합(ordered set)으로 평가되어야만 한다. 첫 번째 요소가 가장 선호되는 인증 문맥 클래스 또는 선언이다. 만약 기술된 클래스들 또는 선언들 중에 어느 것도 아래 규칙들에 따라 충족시키는 것이 없다면, 그러면 응답자는 urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext 를 차상위 수준(second-level) <StatusCode>로 가지는 <Response> 메시지를 반환해야만 한다.

만약 Comparison 이 “exact” 로 설정되어 있거나 또는 생략되어 있으면, 인증 문장에 그 결과로 나타나는 인증 문맥은 명기된 인증 문맥들 중 최소한 하나와는 정확히 일치해야만 한다.

만약 Comparison 이 “minimum” 으로 설정되어 있으면, 인증 문장에 그 결과로 나타나는 인증 문맥은 명기된 인증 문맥들 중 하나와 (응답자가 생각하기에) 적어도 같은 정도의 강도이어야만 한다.

만약 Comparison 이 “better” 로 설정되어 있으면, 인증 문장에 그 결과로 나타나는 인증 문맥은 명기된 인증 문맥들 중 어느 것보다도 (응답자가 생각하기에) 강도가 강해야만 한다.

만약 Comparison 이 “maximum” 으로 설정되어 있으면, 인증 문장에 그 결과로 나타나는 인증 문맥은 명기된 인증 문맥들 중 적어도 하나의 강도를 초과하지 않고 (응답자가 생각하기에) 가능하면 같은 강도이어야만 한다.

다음 스키마 조각은 <RequestedAuthnContext> 요소와 **RequestedAuthnContextType** 복합 타입을 정의한다.

```

    <element name="RequestedAuthnContext"
      type="saml:RequestedAuthnContextType"/>

    <complexType name="RequestedAuthnContextType">
      <choice>

        <element ref="saml:AuthnContextClassRef" maxOccurs="unbounded"/>
        <element ref="saml:AuthnContextDeclRef" maxOccurs="unbounded"/>

      </choice>
      <attribute name="Comparison" type="saml:AuthnContextComparisonType"
        use="optional"/>
    </complexType>

    <simpleType name="AuthnContextComparisonType">
      <restriction base="string">
        <enumeration value="exact"/>
        <enumeration value="minimum"/>
        <enumeration value="maximum"/>
        <enumeration value="better"/>
      </restriction>
    </simpleType>

```

3.3.2.3. <AttributeQuery> 요소

<AttributeQuery> 요소는 “이 주체에 대하여 요구된 속성들을 반환하기” 질의를 하는데 사용된다. 성공적인 응답은 정책이 허용하는 정도까지의 속성 정보들을 포함하는 주장들의 형태가 될 것이다. 이 요소는 **AttributeQueryType** 타입이며, 이것은 다음 요소와 속성을 추가하여 **SubjectQueryAbstractType** 을 확장한다:

<saml:Attribute> [Any Number]

각각의 <saml:Attribute> 요소는 그것의 값(들)이 반환되는 속성을 명기한다. 만약 어떠한 속성들도 명기되지 않는다면, 이것은 정책이 허용하는 모든 속성들이 요구된다는 것을 가리킨다. 만약 주어진 <saml:Attribute> 요소가 하나 또는 그 이상의 <saml:AttributeValue> 요소들을 포함한다면, 그리고 만약 그 속성이 응답에서 반환된다면, 그 속성은 질의에서 명기된 값들과 동일하지 않은 어떠한 값들도 포함하지 않아야만 한다. 특정한 프로파일들 또는 속성들에 의해 동등(equality) 규칙들이 명기되지

않았다면, 동등은 값에 대한 동일한 XML 표현으로 정의된다. <saml:Attribute>에 대한 더 많은 정보는 2.7.3.1 절을 참조한다.

단일 질의(single query)는 동일한 Name 과 NameFormat 값들을 가지는 두 개의 <saml:Attribute> 요소들을 포함하지 않아야만 한다(즉, 주어진 속성은 질의에서 단지 한번만 명명되어야(named)만 한다.).

속성 질의의 응답으로, SAML 기관은 다음과 같이 속성 문장들을 가지는 주장들을 반환한다.

- 질의의 <Subject> 요소와 매칭을 위하여 3.3.4 절에 주어진 규칙들은 반환될 수 있는 주장들을 식별한다.
- 만약, 어떠한 <Attribute> 요소들이 질의에 존재하면, 그들은 위에서 언급한 것처럼, 반환되는 속성들과 선택적으로 반환되는 값들을 한정하고/필터링(constrain/filter)한다.
- 반환되는 속성들과 값들은 또한 응용에 따라 특수한(application-specific) 정책 고려사항에 따라 한정될 수 있다.

차상위 수준 상태 코드들인 urn:oasis:names:tc:SAML:2.0:status:UnknownAttrProfile 과 urn:oasis:names:tc:SAML:2.0:status:InvalidAttrNameOrValue 들이 질의에 있는 속성 또는 값을 해석하는데 있어 문제가 발생하였다는 것을 가리키는데 사용될 수 있다.

다음 스키마 조각은 <AttributeQuery> 요소와 **AttributeQueryType** 복합 타입을 정의한다.

```
<element name="AttributeQuery" type="samlp:AttributeQueryType"/>

<complexType name="AttributeQueryType">
  <complexContent>
    <extension base="samlp:SubjectQueryAbstractType">
      <sequence>
        <element ref="saml:Attribute" minOccurs="0"
          maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

3.3.2.4. <AuthzDecisionQuery> 요소

<AuthzDecisionQuery> 요소는 “이 증거가 주어진다면, 이 주체가 이 자원에 대하여 이러한 동작들을 하는 것이 허락되어야 하는가?” 질의를 하는데 사용된다. 성공적인 응답은 인가 결정 문장들을 포함하는 주장들의 형태가 될 것이다.

주의: <AuthzDecisionQuery> 특징은 향후 별다른 개선 계획이 없는 상태로, SAML 2.0 현재 동결(frozen) 되었다. 추가적인 기능을 요구하는 사용자들은 확장형 접근 제어 마크업 언어(X.1142 를 참조)을 고려하는 것이 좋으며, 이 언어는 개선된 인가 결정 특징들을 제공한다.

이 요소는 **AuthzDecisionQueryType** 타입이며, 이것은 다음 요소와 속성을 추가하여 **SubjectQueryAbstractType** 을 확장한다:

Resource [Required]

인가가 요청되는 대상인 자원을 가리키는 URI 참조.

<saml:Action> [One or More]

인가가 요청되는 대상인 행동들. 이 요소에 대한 더 많은 정보는 2.7.4.2 절을 참조한다.

<saml:Evidence> [Optional]

SAML 기관이 인가 결정을 내릴 때 의지하는 주장 집합. 이 요소에 대한 더 많은 정보는 2.7.4.3 절을 참조한다.

인가 판단 질의에 대한 응답으로, SAML 기관은 다음처럼 인가 판단 문장을 가진 주장들을 반환한다:

- 질의의 <Subject> 요소와 매칭을 위하여 3.3.4 절에 주어진 규칙들은 반환될 수 있는 주장들을 식별한다.

다음 스키마 조각은 <AuthzDecisionQuery> 요소와 **AuthzDecisionQueryType** 복합 타입을 정의한다.

```

<element name="AuthzDecisionQuery" type="samlp:AuthzDecisionQueryType"/>

<complexType name="AuthzDecisionQueryType">
  <complexContent>
    <extension base="samlp:SubjectQueryAbstractType">
      <sequence>
        <element ref="saml:Action" maxOccurs="unbounded"/>
        <element ref="saml:Evidence" minOccurs="0"/>
      </sequence>
      <attribute name="Resource" type="anyURI" use="required"/>
    </extension>
  </complexContent>
</complexType>

```

3.3.3. <Response> 요소

<Response> 요소는 요청을 만족시키는 영 또는 그 이상의 주장들로 응답이 구성될 때 사용된다. 이것은 **ResponseType** 타입이며, 이것은 다음 요소와 속성을 추가하여 **StatusResponseType** 을 확장한다:

<saml:Assertion> 또는 <saml:EncryptedAssertion> [Any Number]

주장을 값으로 기술하거나 또는 선택적으로 암호화된 주장을 값으로 기술한다. 이 요소에 대한 더 많은 정보는 2.3.3 절을 참조한다.

다음 스키마 조각은 <Response> 요소와 **ResponseType** 복합 타입을 정의한다.

```

<element name="Response" type="samlp:ResponseType"/>

<complexType name="ResponseType">
  <complexContent>
    <extension base="samlp:StatusResponseType">
      <choice minOccurs="0" maxOccurs="unbounded">
        <element ref="saml:Assertion"/>
        <element ref="saml:EncryptedAssertion"/>
      </choice>
    </extension>
  </complexContent>
</complexType>

```

3.3.4. 처리 규칙들

SMAL 에서 정의된 질의 메시지에 대한 응답으로, SAML 기관에서 반환하는 모든 주장은 질의에서 발견되는 <saml:Subject> 요소와 강하게 매치(match) 되는 <saml:Subject>를 포함해야만 한다.

<saml:Subject> 요소 S1 이 S2 와 강하게 매치되기 위한 필요 충분조건은 다음의 두가지 조건이 모두 적용되는 것이다.

- 만약 S2 가 하나의 식별자 (<BaseID>, <NameID>, 또는 <EncryptedID>)를 포함한다면, 그러면 S1 은 반드시 동일한 식별자 요소를 포함해야만 한다. 그러나 이 요소는 S1 또는 S2 한 곳에서 암호화되거나 또는 되지 않을 수 있다. 바꾸어 말하면, 식별자의 복호화된 형태는 S1 과 S2 에서 동일해야만 한다. “동일한” (“identical”)이 갖는 의미는 식별자 요소의 내용과 속성 값들이 동일해야만 한다는 것이다. 암호화된 식별자는 일단 복호화되면, 그것의 정의에 따라 원래의 것과 동일하게 될 것이다.
- 만약 S2 가 하나 또는 그 이상의 <saml:SubjectConfirmation> 요소들을 포함하고 있으면, 그러면 S1 는 S2 의 최소한 하나의 <saml:SubjectConfirmation> 요소에서 설명된 방식으로 확인될 수 있는 그러한 <saml:SubjectConfirmation> 요소를 최소한 하나는 포함해야만 한다.

어떤 것이 허용되고 허용되지 않는지에 대한 예로써, S1 은 특정한 Format 값을 갖는 <saml:NameID>를 포함할 수 있고, S2 는 S2 의 <saml:NameID>를 암호화한 결과인 <saml:EncryptedID> 요소를 포함할 수 있다. 그렇지만, 비록 두 개의 식별자들이 동일한 사용자(principal)을 참조한다고 간주되더라도, S1 과 S2 는 다른 Format 값들과 요소 내용을 가지는 <saml:NameID>를 포함할 수는 없다.

만약 SAML 기관이 질의나 또는 주장 참조를 통해 표현된 제한들(constraints)를 만족하는 어떠한 문장들을 가지는 주장을 제공할 수 없다면, <Response> 요소는 <Assertion> 요소를 포함하지 않아야만 되고 urn:oasis:names:tc:SAML:2.0:status:Success 값을 가지는 <StatusCode> 요소를 포함해야만 한다.

기반(underlying) 요청과 응답 메시지들과 관련된 모든 다른 처리 규칙들이 준수되어야만 한다

3.4. 인증 요청 프로토콜 (Authentication Request Protocol)

사용자(principal) (또는 사용자를 대신하여 동작하는 에이전트)가 하나 또는 그 이상의 의지하는 기관들에 보안 문맥(security context)을 설립(establish)하기 위해 인증 문장들을 포함하는 주장들을 얻는 것을 원할 때, 사용자는 <AuthnRequest> 메시지 요소를 SAML 기관에게 전송하고 SAML 기관이 하나 또는 그 이상의 그와 같은 주장들을 포함하는 <Response> 메시지를 반환할 것을 요구하는데 인증 요청 프로토콜을 사용할 수 있다. 이렇게 반환되는 주장들은 어떠한 타입의 추가적인 문장들을 포함할 수도 있지만 그들 중 최소한 하나의 주장은 최소한 하나의 인증 문장을 포함해야만 한다. 이 프로토콜을 지원하는 SAML 기관은 또한 IdP(identity provider, 아이덴티티 제공자)라고 불린다.

이 요구사항과는 별도로, 반환되는 주장들의 특정한 내용들은 사용되는 프로파일 또는 문맥에 따라 결정된다. 또한, 비록 인증 수단이 응답의 내용에 영향을 줄지라도, 사용자 또는 에이전트가 어떤 방식으로 IdP에게 인증되는지는 명기되지 않는다. IdP에 의한 인증 크리덴셜들의 검증 또는 IdP와 인증 처리에 관련된 다른 어떠한 엔티티들 사이의 통신과 관련된 다른 이슈들은 또한 이 프로토콜의 범위를 벗어난다.

다음 절들에서 기술되는 설명들과 처리 규칙들은 다음 행위자들을 참조하며, 이들 중 많은 행위자들은 사용하는 특정 프로파일에서 똑같은 엔티티가 될 지도 모른다:

요청자(Requester)

인증 요청을 생성하고 요청에 대한 응답이 반환되는 엔티티임.

제출자(Presenter)

IdP에게 요청을 제출하는 엔티티임. 제출자는 메시지의 전달 사이에 자신을 인증하거나 또는 자신의 신원을 확인하는데 기존의 보안 문맥을 의지한다. 만약 요청자가 아니면, 제출자는 요청자와 응답하는 IdP 사이에 있는 중개자(intermediary)로서 행동한다.

요청된 주체(Requested Subject)

그것에 대하여 하나 또는 그 이상의 주장들이 요구되고 있는 엔티티.

증명하는 엔티티(Attesting Entity)

결과로 생성되는 주장(들)의 <SubjectConfirmation> 요소들 중에 하나를 만족시킬 수 있을 것으로 기대되는 엔티티 또는 엔티티들.

의지하는 기관(Relying Party)

사용 중인 프로파일 또는 문맥에 의해 정의된 어떤 목적으로 달성하기 위해 주장(들)을 소비할 것으로 예상되는 엔티티 또는 엔티티들. 일반적으로 보안 문맥을 설립할 것으로 기대됨.

IdP(Identity Provider, 아이덴티티 제공자)

제출자가 요청을 전달하고 응답을 수신하는 엔티티.

3.4.1. <AuthnRequest> 요소

IdP 가 인증 문장을 가진 주장을 발급하도록 요청하기 위해, 제출자는 IdP 에게 인증 받고 (또는 이미 존재하는 보안 문맥을 의지하고), IdP 에게 요청의 결과로 생성되는 주장이 그것의 목적을 충족시키기 위해 필요한 특징들(properties)을 설명하는 <AuthnRequest> 메시지를 전달한다. 이러한 특징들 중에는 주장의 내용과 관련된 정보 그리고/또는 결과로 생성되는 <Response> 메시지가 제출자에게 어떻게 배달되어야 하는지와 관련된 정보가 있을 수 있다.

만약, 예를 들어, 요청자가 결과로 생성되는 주장을 요청된 주체를 인증하거나 인가하는데 사용하기를 의도하는 의지하는 기관이라면 그래서 의지하는 기관이 서비스를 제공할 것인지 하지 않을 것인지를 결정할 수 있다면, 요청자는 요청의 제출자와 동일하지 않을 수도 있을 것이다.

<AuthnRequest> 메시지는 서명되어야 하고 만약 그렇지 않다면 메시지를 전달하는 프로토콜 바인딩에 의해 인증되고 무결성이 보호되어야 한다

이 메시지는 **AuthnRequestType** 복합타입이며, 이 타입은 **RequestAbstractType** 을 확장하고 다음 요소들과 속성들을 추가한다. 이들 추가되는 요소들과 속성들은 모두 일반적으로 선택적인 항목이지만 특정 프로파일에서는 요구되는 항목이 될 수도 있다:

<saml:Subject> [Optional]

결과로 생성되는 주장(들)의 요청된 주체를 명기한다. 이것은 하나 또는 그 이상의 <saml:SubjectConfirmation> 요소들을 포함하여 결과로 생성되는 주장들이 어떻게 그리고/또는 누구에 의해 확인될 수 있는지를 지시할 수 있다. 더 많은 정보는 2.4 절을 참조한다.

만약 전체가 생략되거나 또는 식별자가 포함되지 않으면, 메시지 제출자가 요청된 주체로 여겨진다. 만약 <saml:SubjectConfirmation> 요소들이 포함되지 않으면, 제출자는 단지 요구된 증명하는 엔티티로 여겨지고 주체 확인 방법은 사용 중인 프로파일 그리고/또는 IdP 의 정책들에 의해 암묵적으로 결정된다(imply).

<NameIDPolicy> [Optional]

요청된 주체를 나타내는데 사용된 이름 식별자에 대한 제약(constraints)를 명기한다. 만약 생략되면, 요청되는 주체에 대하여 IdP 에서 지원하는 어떠한 타입의 식별자도 사용될 수 있다. 그렇지만 이러한 식별자는 예를 들어, 프라이버시 등과 같은 배치에 따라 특수한(deployment-specific) 정책들에 의해 한정된다.

<saml:Conditions> [Optional]

요청자가 결과로 생성되는 주장(들)의 유효성 그리고/또는 사용을 제약하기 위해 기대하는 SAML 조건들을 명기한다. 응답자는 그것이 필요하다고 생각되면 이 집합을

변경하거나 또는 보완할 수 있다. 이 요소에 있는 정보는 요청 자체의 사용에 대한 조건들으로써가 아니라, 주장을 만드는 과정에 입력(input)으로 사용된다. (더 자세한 정보는 2.5 절을 참조)

<RequestedAuthnContext> [Optional]

만약에 있다면, 제출자에 대한 응답자의 인증에 적용되는 인증 문맥에 대하여 제출자가 부가하는 요구조건들을 기술한다. 이 요소에 관한 처리 규칙은 3.3.2.2.1 절을 참조한다.

<Scoping> [Optional]

요청자가 제출자를 인증할 수 있다고 신뢰하는 IdP 들의 집합을 나타냄. 또한 응답자가 후속(subsequent) IdP 에게 요청자의 <AuthnRequest> 메시지를 대리(proxying)하는 것과 관련된 제약사항들과 문맥을 명기한다.

ForceAuthn [Optional]

부울린 값(boolean value). 만약 “true” 이면, IdP 는 이전의 보안 문맥을 의지하는 대신, 제출자를 직접적으로 인증해야만 한다. 만약 값이 제공되지 않는다면, 기본은 “false” 이다. 그러나, 만약 ForceAuthn 과 IsPassive 가 둘 다 “true”이면, IsPassive 제약을 충족시킬수 없다면 IdP 는 제출자를 새롭게 인증한지 않아야만 된다.

IsPassive [Optional]

부울린 값(boolean value). 만약 “true” 이면, IdP 와 사용자 에이전트는 요청자로부터 눈에 보이게 사용자 인터페이스를 제어하지 않아야만 하며 눈에 띄는(noticeable) 방식으로 제출자와 상호작용(interaction)을 하지 않아야만 한다. 만약 값이 제공되지 않는다면, 기본은 “false” 이다.

AssertionConsumerServiceIndex [Optional]

<Response> 메시지가 요청자에게 반환되어야 하는 위치를 간접적으로 식별한다. 이것은 SAML 2.0 프로파일 표준의 웹 브라우저 SSO 프로파일과 같이 요청자가 제출자와 다른 프로파일들에만 적용된다. IdP 는 이 속성의 인덱스 값과 요청자와 관련된 위치를 매핑시키는 신뢰되는 수단을 가지고 있어야만 한다. SAML 2.0 메타데이터 표준은 하나의 가능한 메커니즘을 제공한다. 만약 생략되면, IdP 는 사용중인 프로파일에 대하여 요청자와 관련된 기본 위치로 <Response> 메시지를 반환해야만 한다. 만약 기술된 인덱스가 유효하지 않으면, IdP 는 에러 <Response>를 반환하거나 또는 기본 위치를 사용할 수 있다. 이 속성은 AssertionConsumerServiceURL 과 ProtocolBinding 속성들과는 상호 배타적이다.

AssertionConsumerServiceURL [Optional]

<Response> 메시지가 요청자에게 반환되어야 하는 위치를 값으로 명기한다. 응답자는 어떻게 해서든지 명기된 값이 실제로 요청자와 관계가 있다는 것을 확인해야만 한다. SAML 2.0 메타데이터 표준은 하나의 가능한 메커니즘을 제공한다; 둘러싸는

<AuthnRequest> 메시지를 서명하는 것은 또 다른 가능한 메커니즘이다. 이 속성은 AssertionConsumerServiceIndex 속성과는 상호 배타적이고 전형적으로 ProtocolBinding 속성을 수반한다.

ProtocolBinding [Optional]

<Response> 메시지를 반환할 때 사용되는 SAML 프로토콜 바인딩을 식별하는 URI 참조. 프로토콜 바인딩들과 그들에 대하여 정의된 URI 참조들에 대한 더 많은 정보는 SAML 2.0 바인딩 표준을 참조한다. 이 속성은 AssertionConsumerServiceIndex 속성과는 상호 배타적이고 전형적으로 AssertionConsumerServiceURL 속성을 수반한다.

AttributeConsumingServiceIndex [Optional]

요청자와 관련이 있으며, 요청자가 IdP 에 의해 <Response> 메시지에 제공되기를 원하거나 또는 요청하는 SAML 속성들을 설명하는 정보를 간접적으로 식별한다. IdP 는 이 속성의 인덱스 값과 요청자와 관련된 정보를 매핑시키는 신뢰되는 수단을 가지고 있어야만 한다. SAML 2.0 메타데이터 표준은 하나의 가능한 메커니즘을 제공한다. IdP 는 그것이 반환하는 주장(들)에 하나 또는 그 이상의 <saml:AttributeStatement> 요소들을 전파시키기(populate) 위해 이 정보를 사용할 수 있다.

ProviderName [Optional]

제출자의 사용자 에이전트 또는 IdP 에 의해 사용될 목적으로 사람이 읽을 수 있는 요청자의 이름을 명기한다.

이 메시지와 관련된 일반적인 처리 규칙들은 3.4.1.4 절을 참조한다.

다음 스키마 조각은 <AuthnRequest>와 그것의 타입인 AuthnRequestType 복합 타입을 정의한다.

```
<element name="AuthnRequest" type="samlp:AuthnRequestType"/>

<complexType name="AuthnRequestType">
  <complexContent>
    <extension base="samlp:RequestAbstractType">
      <sequence>
        <element ref="saml:Subject" minOccurs="0"/>
        <element ref="samlp:NameIDPolicy" minOccurs="0"/>
        <element ref="saml:Conditions" minOccurs="0"/>
        <element ref="samlp:RequestedAuthnContext" minOccurs="0"/>
        <element ref="samlp:Scoping" minOccurs="0"/>
      </sequence>
      <attribute name="ForceAuthn" type="boolean" use="optional"/>
      <attribute name="IsPassive" type="boolean" use="optional"/>
    </extension>
  </complexContent>
</complexType>
```



```

<attribute name="ProtocolBinding" type="anyURI" use="optional"/>
    <attribute name="AssertionConsumerServiceIndex"
        type="unsignedShort"
        use="optional"/>
    <attribute name="AssertionConsumerServiceURL" type="anyURI"
        use="optional"/>
    <attribute name="AttributeConsumingServiceIndex"
        type="unsignedShort" use="optional"/>
    <attribute name="ProviderName" type="string" use="optional"/>
    </extension>
</complexContent>
</complexType>

```

3.4.1.1. <NameIDPolicy> 요소

<NameIDPolicy> 요소는 <AuthnRequest> 로부터 생성된 주장들의 주체들에 있는 이름 식별자를 재단한다 (tailors). 이것의 **NameIDPolicyType** 복합 타입은 다음 속성들을 정의한다:

Format [Optional]

이 표준 또는 다른 표준에서 정의된 이름 식별자 포맷과 대응되는 URI 참조를 명기한다 (예들을 참고하려면 8.3 절을 참조). 추가적인 값인 `urn:oasis:names:tc:SAML:2.0:nameid-format:encrypted` 이 명시적으로 정의된다. 이 값은 요청을 처리한 결과로 생성되는 식별자가 암호화되어야 한다는 것을 지시하기 위해 이 속성에서 사용된다.

SPNameQualifier [Optional]

주장 주체의 식별자가 요청자가 아닌 서비스 제공자의 네임스페이스나 또는 서비스 제공자들의 제휴그룹(affiliation group)의 네임스페이스로 반환 (또는 생성)되어야 한다는 것을 명기하는 선택적인 요소이다. 예를 들어 `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent` 의 정의는 8.3.7 절을 참조한다.

AllowCreate [Optional]

요청을 처리하는 중간에, IdP 가 사용자를 나타내는 새로운 식별자를 생성하는 것이 허용되는지 또는 허용되지 않는지를 지시하는데 사용되는 부울린 값. 기본은 “false”이다. “false”일 때는, 만약 사용자를 위해 수용가능한(acceptable) 식별자가 이미 설정되어 있다면, 요청자는 IdP 가 요청자에게 단지 주장을 발급하기만 하도록 제한한다. 이렇게

하는 것이 IdP 가 (예를 들어, 많은 수의 사용자들에 대하여 미리) 이러한 명기된 요청의 문맥 밖에서 이와 같은 식별자들을 생성하는 것을 방지하지는 않는다는 것에 주의해야 한다.

이 요소가 사용될 때, 만약 IdP 가 요소의 내용을 이해하지 못하거나 또는 받아들이지 못하면, <Response> 메시지 요소는 에러 <status>를 가지고 반환되어야만 한다 그리고 urn:oasis:names:tc:SAML:2.0:status:InvalidNameIDPolicy 를 갖는 차상위 수준 <StatusCode> 포함할 수 있다.

만약 Format 값이 생략되거나 urn:oasis:names:tc:SAML:2.0:nameid-format:unspecified 로 설정되어 있으면, IdP 는 어떠한 종류의 식별자도 자유롭게 반환할 수 있다. 이 때, 이들 식별자는 이 요소의 내용 또는 IdP 나 사용자의 정책들에 기인하는 어떠한 추가적인 제약을 받는다.

특별한 Format 값인 urn:oasis:names:tc:SAML:2.0:nameid-format:encrypted 는 결과로 생성되는 주장들이 평문 대신 <EncryptedID> 요소들을 포함해야만 한다는 것을 가리킨다. 요청된 주체에 대하여 기반이 되는 이름 식별자의 비암호화된 형태는 IdP 에 의해 지원되는 어떠한 형태도 될 수 있다.

<NameIDPolicy>의 Format 에 관계없이, 만약 (가능하면 서비스 제공자에 특수한) IdP 에 유효한 정책들이 암호화된 식별자들이 사용될 것을 요구하면, IdP 는 결과로 생성되는 주장 주체에 <EncryptedID>를 반환할 수 있다

만약 요청자가, 사용자에 대한 식별자가 존재하지 않을 때 IdP 가 사용자에 대한 새로운 식별자를 설정하는 것을 허용할 것을 원한다면, 요청자는 “true”로 설정된 AllowCreate 속성을 가지는 이 요소를 포함해야만 한다. 만약 그렇지 않다면, IdP 가 요청자에 의해 사용할 수 있는 식별자를 이전에 설정해 놓은 사용자만 성공적으로 인증될 수 있다. 이것은 urn:oasis:names:tc:SAML:2.0:nameid-format:persistent Format 값과 결합할 때 주로 유용하다 (8.3.7 절 참조).

다음 스키마 조각은 <NameIDPolicy> 요소와 **NameIDPolicyType** 복합 타입을 정의한다.

```
<element name="NameIDPolicy" type="samlp:NameIDPolicyType"/>

<complexType name="NameIDPolicyType">
  <attribute name="Format" type="anyURI" use="optional"/>
  <attribute name="SPNameQualifier" type="string" use="optional"/>
  <attribute name="AllowCreate" type="boolean" use="optional"/>
</complexType>
```

3.4.1.2. <Scoping> 요소

요청자가 제출자를 인증할 수 있다고 신뢰하는 IdP 들의 집합을 명기한다. 또한 응답자가 후속(subsequent) IdP 에게 요청자의 <AuthnRequest> 메시지를 대리(proxying)하는 것과 관련된 제약사항들과 문맥을 명기한다. 이 요소의 ScopingType 복합 타입은 다음 요소들과 속성들을 정의한다.

ProxyCount [Optional]

이 <AuthnRequest>를 수신하는 IdP 와 최종적으로 사용자를 인증하는 IdP 사이에서 가능한 대리 우회(proxying indirection)의 회수를 기술한다. 만약 0 을 값으로 가지면 대리를 허용하지 않는다는 것을 의미하고, 반면 이 속성을 생략하는 것은 이와 같은 제약을 두지 않겠다는 것을 표현한다.

<IDPLIST> [Optional]

요청자가 이 요청에 응답하는 것이 받아들여질 수 있다고 생각하는 IdP 들과 관련 정보의 리스트. 이 정보는 권고적인(advisory) 정보이다.

<RequesterID> [Zero or More]

요청자를 대신하여 행동하는 요청 엔티티들의 집합을 식별한다. 3.4.1.5 절에서 설명되는 것처럼 대리가 발생할 때, 요청자들의 체인(chain)을 통신하는데 사용된다. 엔티티 식별자들에 대한 설명은 8.3.6 절을 참조한다.

활동적인 중개자(active intermediary)를 명기하는 프로파일들에서, 만약 중개자가 명술된 IdP 들 중에 어떠한 것도 접촉할 수 없거나 또는 지원하지 않는다면, 에러 <Status>와 urn:oasis:names:tc:SAML:2.0:status:NoAvailableIDP 또는 urn:oasis:names:tc:SAML:2.0:status:NoSupportedIDP 을 값으로 갖는 차상위 수준 <StatusCode>를 가지는 <Response> 메시지를 반환할 수 있다.

다음 스키마 조각은 <Scoping> 요소와 **ScopingType** 복합 타입을 정의한다.

```
<element name="Scoping" type="samlp:ScopingType"/>
  <complexType name="ScopingType">
    <sequence>
      <element ref="samlp:IDPLIST" minOccurs="0"/>
      <element ref="samlp:RequesterID" minOccurs="0"
        maxOccurs="unbounded"/>
    </sequence>
    <attribute name="ProxyCount" type="nonNegativeInteger" use="optional"/>
  </complexType>
  <element name="RequesterID" type="anyURI"/>
```

3.4.1.3. <IDPList> 요소

<IDPList> 요소는 요청자에 의해 신뢰되며 제출자를 인증하는 IdP 들을 명기한다. 이것의 IDPListType 복합 타입은 다음 요소들을 정의한다:

<IDPEntry> [One or More]

단일 IdP 에 대한 정보

<GetComplete> [Optional]

만약 <IDPList>가 완전(complete) 하지 않다면, 이 요소를 사용하여 완전한 리스트(complete list)를 검색하는데 사용될 수 있는 URI 참조를 기술한다. URI 와 자원을 검색하는 것은 그 자체로 <GetComplete> 요소를 포함하지 않는 <IDPList>가 루트 요소(root element)인 XML 인스턴스(XML instance)를 검색해와야만 한다.

다음 스키마 조각은 <IDPList> 요소와 IDPListType 복합 타입을 정의한다.

```
<element name="IDPList" type="samlp:IDPListType"/>
```

```
<complexType name="IDPListType">
```

```
<sequence>
```

```
<element ref="samlp:IDPEntry" maxOccurs="unbounded"/>
```

```
<element ref="samlp:GetComplete" minOccurs="0"/>
```

```
</sequence>
```

```
</complexType>
```

```
<element name="GetComplete" type="anyURI"/>
```

<IDPEntry> 요소는 요청자에 의해 신뢰되며 제출자를 인증하는 단일한 IdP 를 명기한다. 이것의 IDPEntryType 복합 타입은 다음 속성들을 정의한다.

ProviderID [Required]

IdP 의 유일한 식별자. 이와 같은 식별자들에 대한 설명은 8.3.6 절을 참조한다.

Name [Optional]

사람이 읽기 편한 IdP 이름.

Loc [Optional]

인증 요청 프로토콜을 지원하는 프로토콜에 따라 고유한(protocol-specific) 엔드포인트의 위치를 나타내는 URI 참조. 사용되는 바인딩은 사용중인 프로파일로부터 이해되어야만 한다.

다음 스키마 조각은 <IDPEntry> 요소와 IDPEntryType 복합 타입을 정의한다.

```

<element name="IDPEntry" type="samlp:IDPEntryType"/>

<complexType name="IDPEntryType">
  <attribute name="ProviderID" type="anyURI" use="required"/>
  <attribute name="Name" type="string" use="optional"/>
  <attribute name="Loc" type="anyURI" use="optional"/>
</complexType>

```

3.4.1.4. 처리 규칙들

<AuthnRequest>와 <Response> 교환은 다양한 사용 시나리오들을 지원함에 따라, 일반적으로 이러한 선택성이 제약되고 특정한 종류의 입력과 출력이 요구되거나 금지되는 특정 환경에서 사용될 수 있도록 프로파일화된다(profiled). 다음 처리 규칙은 이 프로토콜 교환을 사용하는 어떠한 프로파일에서도 항상 적용되는 행동을 나타낸다. 기반 요청과 응답 메시지들과 연계된 모든 다른 처리 규칙들 또한 반드시 준수되어야만 한다.

응답자는 최종적으로 요청에 의해 정의된 사항들을 충족하는 하나 또는 그 이상의 주장들을 포함하는 하나의 <Response> 메시지 또는 발생한 에러를 설명하는 하나의 <Status>를 포함하는 하나의 <Response> 메시지를 가지고 <AuthnRequest>에 응답해야만 한다. 프로토콜 바인딩과 인증 메커니즘의 특성에 영향을 받아, 인증 과정을 시작하거나 완료하기 위해 필요할 때, 응답자는 제출자와 추가적인 메시지 교환을 수행할 수 있다. 다음 절에서 설명하는 것처럼, 이것은 IdP 가 <AuthnRequest> 발급하여 제출자를 또 다른 IdP 에게 위치시킴(directing)으로써 요청을 대리하는 것을 포함하고, 이렇게 함으로써 결과로 생성되는 주장은 인증 메커니즘으로써 사실상 SAML 을 사용하여 제출자를 원래의 응답자에게 인증시키는데 사용될 수 있다.

만약 응답자가 제출자를 인증할 수 없거나 요청된 주체를 인식하지 못하면, 또는 만약 응답자가 IdP 에 효력이 있는 정책들에 의해 주장을 제공하는 것이 금지되면 (예를 들어, 의도된 주체가 IdP 가 의지하는 기관에게 주장을 제공하는 것을 금지했다), 그러면 응답자는 에러 <Status>와 urn:oasis:names:tc:SAML:2.0:status:AuthnFailed 또는 urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal 을 값으로 갖는 차상위 수준 <StatusCode>를 가지는 <Response> 메시지를 반환해야만 한다.

만약 요청에 <saml:Subject> 요소가 존재하면, 그러면 결과로 생성되는 주장들의 <saml:Subject>는 3.3.4 절에서 설명된 것과 같이 요청 <saml:Subject>와 강하게 매치되어야만 한다. 이 경우, 만약 식별자가 <NameIDPolicy>에 의해 명기되었다면, 식별자가 다른 포맷일 수 있다. 이와 같은 경우, 식별자의 물리적인 내용이 다를 수 있지만이것은 동일한 사용자를 참조해야만 한다.

비록 일부는 어떤 프로파일들에서 필수요소이지만, <AuthnRequest> 내에 명시적으로 정의된 모든 내용은 선택적인 항목이다. 어떠한 명기된 내용이 없는 경우, 다음과 같은 행동이 암시적으로 수행된다.

- 반환되는 주장(들)은 제출자를 나타내는 하나의 <saml:Subject> 요소를 포함해야만 한다. 식별자 타입과 포맷은 IdP 에 의해 결정된다. 적어도 하나의 주장에서 최소한 하나의 문장은 응답자 또는 응답자와 관련된 인증 서비스에 의해 수행된 인증을 설명하는 <saml:AuthnStatement> 이어야만 한다.
- 가능한 정도까지, 요청 제출자는 주장(들)의 <saml:SubjectConfirmation>을 충족시킬수 있는 유일한 증명하는 엔티티이어야 한다. 더 약한 확인 방법들의 경우, 바인딩에 따라 고유한 메커니즘 또는 다른 메커니즘들이 이러한 요구사항을 충족시키는데 사용될 것이다.
- 결과로 생성되는 주장(들)은 받아들일 수 있는 의지하는 기관으로 요청자를 참조하는 <saml:AudienceRestriction> 요소를 포함해야만 한다. 다른 청중들은 IdP 에 의해 적절하다고 생각될 때 포함할 수 있다

3.4.1.5. 대리(Proxying)

만약, <AuthnRequest>를 수신한 IdP 가 아직 제출자를 인증하지 못했거나 또는 직접적으로 제출자를 인증할 수 없지만, 그러나 제출자가 이미 다른 IdP 나 또는 SAML 이 아닌 제공자(non-SAML equivalent)에게 인증되었다고 믿는다면, IdP 는 다른 IdP 에게 제출하는 새로운 <AuthnRequest>를 발급하거나 그 엔티티가 인식하는 SAML 이 아닌 포맷으로된 요청을 발급함으로써 원래의 요청에 대하여 응답할 수 있다. 원래의 IdP 는 대리하는 IdP 로 불린다.

대리하는 제공자에게 <Response> (또는 SAML 이 아니면서 <Response>와 동등한 것)를 성공적으로 반환하면, 반환된 응답에 포함된 주장 또는 SAML 이 아니지만 동등한 것이 제출자를 인증하는데 사용될 수 있다. 이렇게 함으로써 대리하는 제공자는 자신이 받은 최초의 <AuthnRequest>에 대한 응답으로 자신의 주장을 발급할 수 있다. 이와 같은 과정을 통해 전체 메시지 교환을 완료한다. 앞 절과 아래에서 설명되는 것처럼, 대리하는 IdP 와 인증하는 IdP 는 둘 다 대리 행동에 대한 제한사항을 그들이 발행하는 메시지들과 주장들에 포함시킬 수 있다.

요청자는 제공자가 바람직한 ProxyCount 값을 설정할 수 있는 <Scoping> 요소를 포함하고 그리고/또는, 선호되는 제공자들의 순서를 가진 <IDPList>를 포함함으로써 대리될 수 있는 선호되는 IdP 들의 리스트를 지시할 수 있다. 이와 같은 행동들은 대리 행동에 영향을 줄 수 있다.

IdP 는 그것이 발급하는 주장들의 <ProxyRestriction>을 이용하여 자신의 주장들이 대리 IdP 들에 의해 이차적으로(secondary) 어떻게 사용되는지를 제어할 수 있다.

만약 <ProxyCount> 속성이 생략되거나 또는 0 보다 크면, IdP 는 <AuthnRequest>를 대리할 수 있다. IdP 가 대리할지 말지를 선택하는 것은 IdP 의 지역 정책 문제이다. 만약 이 속성이 제공된다면, IdP 는 <IDPList>에 명기된 제공자를 대리할 수 있지만 반드시 그렇게 할 필요는 없다.

IdP 는 <ProxyCount>가 0 으로 설정되어 있는 요청을 대리하지 않아야만 한다. 만약 IdP 가 제출자를 직접 인증할 수 없다면, IdP 는 urn:oasis:names:tc:SAML:2.0:status:ProxyCountExceeded 를 차상위 수준 <StatusCode> 값으로 가지는 예러 <Status>를 반환해야만 한다.

만약, 대리하는 IdP 가 SAML IdP 에게 대리하기로 한다면, 새로운 <AuthnRequest>를 생성할 때, 대리하는 IdP 는 원래 요청에 포함된 모든 정보에 대하여 동일하거나 또는 더 엄격한 형태의 정보를 포함해야만 한다. 그러나 대리하는 제공자는 그것이 성공적인 응답 기회를 최대화하기 위하여 어떠한 <NameIDPolicy>를 기술해도 좋다는 것에 주의해야 한다.

만약 인증하는 IdP 가 SAML IdP 가 아니면, 대리하는 제공자는 (예를 들어 <IsPassive>와 같이) 사용자 에이전트 상호작용을 지배하는 요소들이 인증하는 제공자에서 존중될(honor) 것임을 어떠한 방식으로든 보장해야만 한다.

새로운 <AuthnRequest>는 원래 값보다 기껏해야 1 적은 값을 가지는 <ProxyCount> 속성을 포함해야만 한다. 만약 원래 요구가 <ProxyCount> 속성을 포함하지 않으면, 새로운 요청은 <ProxyCount> 속성을 포함해야 한다.

만약 <IDPList>가 원래 요청에 명기되지 않았다면, 새로운 요청은 <IDPList>를 포함해야만 한다. 대리하는 IdP 는 추가적인 IdP 들을 <IDPList>의 끝에 추가할 수 있지만 <IDPList>에 있는 것을 삭제하지 않아야만 한다.

인증 요청과 응답은 이 절에서 주어진 규칙들과 사용 중인 프로파일에 따라 정규(normal) 방식으로 처리된다. (SAML 의 경우 <Response>를 배달함으로써) 일단 제출자가 대리하는 IdP 에게 인증되었으면, 다음 절차들이 준용된다:

- 대리하는 IdP 는 원래의 주장 또는 SAML 이 아닌 동치로부터 관련된 정보를 복사함으로써 자신의 새로운 주장을 생성할 것을 준비한다.
- 새로운 주장의 <saml:Subject>는 그것의 <NameIDPolicy> 요소로 정의된 대로, 원래 요청자의 선호들(preferences)을 만족시키는 식별자를 포함해야만 한다.
- 새로운 <saml:AuthnStatement>는 대리하는 IdP 가 제출자를 인증하도록 요청한 IdP 를 참조하는 <saml:AuthenticatingAuthority>를 포함하는 <saml:AuthnContext>를 포함해야만 한다. 만약 응답에 있는 주장이 하나 또는 그 이상의 <saml:AuthenticatingAuthority> 요소들을 포함하는 <saml:AuthnContext> 정보를 포함하면, 새로운 <saml:AuthenticatingAuthority> 요소가 그들 뒤에 위치시키도록 하면서 새로운 주장에 포함되어야 한다.
- 만약 인증하는 IdP 가 SAML 제공자가 아니면, 그러면 대리하는 IdP 는 인증하는 제공자에 대한 유일한 식별자 값을 생성해야만 한다. 이 값은 다른 요청들 상에서 시간이 지나도 일관성이 있어야 한다. 이 값은 다른 SAML 제공자들에 의해 사용되거나 또는 생성된 값들과 충돌이 발생하지 않아야만 한다.
- 만약 요청자가 구술한(dictate) 원래의 요구사항들이 충족되어진다면, 어떠한 다른<saml:AuthnContext> 정보도 대리하는 IdP 정책들에 따라 복사되거나, 번역되거나(translated) 또는 생략 수 있다.

만약, 미래에 또 다른 요청을 통해 동일한 제출자를 인증하도록 IdP 가 요구받고 이 요청이 (대리하는 IdP 에 의해 판단될 때) 원래 요청과 동등하거나 또는 덜 엄격(strict)하면, IdP 는 인증하는 IdP 에게 전달할 새로운 <AuthnRequest>를 생성하는 것을 건너뛰고 즉시 다른 주장을 발급할 수 있다(단 IdP 가 수신했던 원 주장이나 비-SAML 동치가 여전히 유효하다는 것이 가정되어야 한다).

3.5. 아티팩트 해결 프로토콜(Artifact Resolution Protocol)

아티팩트(artifact) 해결 프로토콜은 SAML 메시지들이 값에 의해서가 아니라 참조에 의해서 SAML 바인딩에 전달될 수 있게 하는 메커니즘을 제공한다. 이 특별한 프로토콜을 사용하여 요청들과 응답들 둘 다 모두 참조에 의해 얻어질 수 있다. 메시지를 트랜스포터 프로토콜로 바인딩하는 대신에, 메시지 송신자는 이 바인딩을 사용하여 아티팩트로 불리는 작은 조각의 데이터를 송신한다. 아티팩트는 다양한 형태들을 가질 수 있지만 반드시 수신자가 그것을 보낸 송신자가 누구인지를 판단할 수 있는 수단들을 지원해야만 한다. 만약 수신자가 원하면, 수신자는 아티팩트를 가지고 원래의 프로토콜 메시지를 얻기 위해 이 프로토콜을 다른 (일반적으로 동기적인(synchronous)) SAML 바인딩 프로토콜과 함께 사용할 수 있다.

크기 제약 때문에 메시지를 쉽게 운반(carry)할 수 없거나 또는 서명에 대한 요구를 피하면서, SAML 요청자와 응답자 사이에서 안전한 채널을 통해 메시지가 통신될 수 있도록 하는 바인딩들에서 이 메커니즘이 가장 일반적으로 사용된다.

기반하는 메시지가 참조에 의해 전달된다는 특성에 의존하기 때문에, 아티팩트 해결 프로토콜은 아티팩트를 해결하는데 사용되는 프로토콜 바인딩으로부터 상호 인증, 무결성 보호, 기밀성, 등등과 같은 보호들을 요구할 수 있다. 모든 경우에, 아티팩트는 단일-사용 의미(single-use semantic)를 보여(exhibit)야만 한다. 단일-사용 의미는 일단 아티팩트가 성공적으로 해결되면, 그 아티팩트는 어떠한 기관에서도 더 이상 재 사용될 수 없다는 것을 의미한다.

얻어진 프로토콜 메시지에 관계없이, 아티팩트 해결의 결과는 그렇게 얻어진 메시지가 아티팩트 대신 원래 보내졌던 것처럼 처리되어야만 한다

3.5.1. <ArtifactResolve> 요소

<ArtifactResolve> 메시지는 SAML 프로토콜 메시지를 나타내는 아티팩트를 명기함으로써 SAML 프로토콜 메시지가 <ArtifactResponse> 메시지에 포함되어 반환되도록 요청하는데 사용된다. 아티팩트의 최초 전달은 사용되고 있는 특정 프로토콜 바인딩에 따라 결정된다. 바인딩들에서 아티팩트들을 어떻게 사용하는지에 대한 정보는 SAML 2.0 바인딩 표준을 참조한다.

<ArtifactResolve> 메시지는 서명되거나 또는 그렇지 않다면 메시지를 전달하는데 사용되는 프로토콜 바인딩에 의해 인증되고 무결성을 보장받아야 한다.

이 메시지는 **ArtifactResolveType** 복합타입이며, 이 타입은 **RequestAbstractType** 을 확장하고 다음 요소들과 속성들을 추가한다:

<Artifact> [Required]

요청자가 받았고 그리고 지금은 그것이 나타내는 프로토콜 메시지로 변환하기를 원하는 아티팩트 값.

다음 스키마 조각은 <ArtifactResolve> 요소와 **ArtifactResolveType** 복합 타입을 정의한다.

```

<element name="ArtifactResolve" type="samlp:ArtifactResolveType"/>

<complexType name="ArtifactResolveType">
  <complexContent>
    <extension base="samlp:RequestAbstractType">
      <sequence>
        <element ref="samlp:Artifact"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<element name="Artifact" type="string"/>

```

3.5.2. <ArtifactResponse> 요소

<ArtifactResolver> 메시지의 수신자는 <ArtifactResponse> 메시지 요소를 가지고 응답해야만 한다. 이 요소는 **ArtifactResponseType** 복합 타입이며, 이것은 반환되는 SAML 프로토콜 메시지에 대응되는 단일한 선택적인 와일드 카드(wildcard) 요소를 가지고 **StatusResponseType**를 확장한다.

<ArtifactResponse> 메시지는 서명되거나 또는 그렇지 않다면 메시지를 전달하는데 사용되는 프로토콜 바인딩에 의해 인증되고 무결성을 보장받아야 한다.

다음 스키마 조각은 <ArtifactResponse> 요소와 **ArtifactResponseType** 복합 타입을 정의한다.

```

<element name="ArtifactResponse" type="samlp:ArtifactResponseType"/>

<complexType name="ArtifactResponseType">
  <complexContent>
    <extension base="samlp:StatusResponseType">
      <sequence>
        <any namespace="##any" processContents="lax"
          minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

3.5.3. 처리 규칙들

만약 응답자가 아티팩트를 유효한 것으로 인식하면, 응답자는 <ArtifactResponse> 메시지 요소에 관련된 프로토콜 메시지를 포함하여 응답한다. 만약 그렇지 않다면, 응답자는 어떠한 내장된(embedded) 요소도 가지지 않는 <ArtifactResponse>를 가지고 응답한다. 두 가지 경우 모두, <Status> 요소는 urn:oasis:names:tc:SAML:2.0:status:Success 를 코드 값으로 가지는 <StatusCode>를 포함해야만 한다. 그것 안에 어떠한 내장된 메시지도 가지고 있지 않은 응답 메시지는 이 절의 나머지 부분에서 빈 응답으로 불린다.

응답자는 임의의 요청자가 동일한 아티팩트를 사용하는 어떠한 후속(subsequent) 요청도 위에서 설명한 대로 빈 응답을 초래한다는 것을 보장함으로써 아티팩트에 일회성-사용(one-time-use) 특성을 실시(enforce)해야만 한다.

일부 프로파일들에서 가장 흔히 <AuthnRequest> 메시지인 경우가 많은, 일부 SAML 프로토콜 메시지들은 그것을 수신하고 적절하게 응답할 수 있는 어떠한 기관에 의해서도 소비될 것으로 의도될 수 있다. 그러나 대부분의 다른 경우에는, 메시지는 특정한 엔티티를 위한 것으로 의도된다. 이와 같은 경우에, 아티팩트를 발급할 때, 아티팩트는 그것이 나타내는 메시지에 대한 의도된 수신자와 연계되어야만 한다. 만약 아티팩트 발급자가 그 자신이 원래 의도된 수신자임을 인증할 수 없는 요청자로부터 <ArtifactResolve> 메시지를 수신하면, 아티팩트 발급자는 빈 응답을 반환해야만 한다.

받아들일 수는 있으나 그 이상을 초과하지 않는 시각 윈도우(window of time)가 아티팩트를 얻는 아티팩트 수신자에게 존재하고 그리고 <ArtifactResolve> 메시지에 아티팩트를 담아 발급자에게 반환할 수 있도록, 아티팩트 발급자는 아티팩트의 유용성(usability)에 가장 짧은 실용적인 시각 제약을 실시해야 한다.

<ArtifactResponse> 메시지의 InResponseTo 속성은 대응되는 <ArtifactResolve> 메시지의 ID 속성 값을 포함해야만 한다는 것에 주의해야 한다. 그러나 내장된 프로토콜 메시지는 그 자신의 메시지 식별자를 가질 것이고, 내장된 응답의 경우에는 내장된 메시지가 응답하고 있는 원래의 요청 메시지와 대응되는 다른 InResponseTo 값을 가질 수도 있다.

기반 요청과 응답 메시지들과 관련된 모든 다른 처리 규칙들은 준수되어야만 한다

3.6. 이름 식별자 관리 프로토콜(Name Identifier Management Protocol)

사용자(principal)를 위한 이름 식별자를 설정한 후, 사용자를 참조할 때 자신이 사용할 식별자의 값과/또는 포맷을 변경하길 원하거나, 또는 이름 식별자가 더 이상 그 사용자를 참조하는데 사용되지 않을 것이라는 사실을 가리키기를 원하는 IdP 는 서비스 제공자들에게 <ManageNameIDRequest> 메시지를 송신함으로써 이와 같은 변화를 알린다.

기반(underlying) 이름 식별자가 자신과 통신하는데 사용될 때, 서비스 제공자 또한 이 메시지를 이용하여 SPProviderID 값을 등록하거나 변경한다. 또한 서비스 제공자는 자신과 IdP 간의 이름 식별자의 사용을 종료시키려 할 때 이 메시지를 사용한다.

이 프로토콜은 전형적으로 일시적인(transient) 이름 식별자들을 가지고 사용되지 않는다. 이것은 일시적인 이름 식별자들의 값이 긴 기간(long term) 관리될 목적을 가지고 있지 않기 때문이다.

3.6.1. <ManageNameIDRequest> 요소

제공자는 수신자에게 변화된 이름 식별자를 알리거나 또는 이름 식별자 사용의 종료를 가리키기 위해 <ManageNameIDRequest> 메시지를 전달한다.

<ManageNameIDRequest> 메시지는 서명되거나 또는 그렇지 않다면 메시지를 전달하는데 사용되는 프로토콜 바인딩에 의해 인증되고 무결성을 보장받아야 한다.

이 메시지는 **ManageNameIDRequestType** 복합 타입을 가지며, 이것은 **RequestAbstractType** 을 확장하며 다음 요소들을 추가한다.

<saml:NameID> 또는 <saml:EncryptedID> [Required]

이 요청을 하기 이전에 IdP 와 서비스 제공자들에 의해 현재 인지되고 있는 형태로 사용자를 명기하는 (평문 또는 암호화 형태인) 이름 식별자와 관련된 설명 데이터. (이 요소에 대한 더 많은 정보는 2.2 절을 참조)

<NewID> 또는 <NewEncryptedID> 또는 <Terminate> [Required]

이 사용자에게 대하여 요청하는 제공자와 통신할 때 사용될 (평문 또는 암호화 형태인) 새로운 이름 식별자 값, 또는 이전(old) 식별자의 사용이 종료되었음을 가리키는 값. 전자의 경우, 만약 요청자가 서비스 제공자이면, 새로운 식별자는 후속(subsequent) <NameID> 요소들에 SPProvidedID 속성으로 나타나야만 한다. 만약 요청자가 IdP 이면, 새로운 값은 뒤따르는 <NameID> 요소들에서 요소의 값으로 나타날 것이다.

다음 스키마 조각은 <ManageNameIDRequest> 요소와 **ManageNameIDRequestType** 복합 타입을 정의한다.

```

    <element name="ManageNameIDRequest"
    type="samlp:ManageNameIDRequestType"/>

    <complexType name="ManageNameIDRequestType">
        <complexContent>
            <extension base="samlp:RequestAbstractType">
                <sequence>
                    <choice>
                        <element ref="saml:NameID"/>
                        <element ref="saml:EncryptedID"/>
                    </choice>
                    <choice>
                        <element ref="samlp:NewID"/>
                        <element ref="samlp:NewEncryptedID"/>
                        <element ref="samlp:Terminate"/>
                    </choice>
                </sequence>
            </extension>
        </complexContent>
    </complexType>

    <element name="NewID" type="string"/>
    <element name="NewEncryptedID" type="saml:EncryptedElementType"/>
    <element name="Terminate" type="samlp:TerminateType"/>

    <complexType name="TerminateType"/>

```

3.6.2. <ManageNameIDResponse> 요소

<ManageNameIDRequest> 메시지의 수신자는 <ManageNameIDResponse> 메시지를 가지고 응답해야만 한다. 이 요소는 추가적인 내용을 가지고 있지 않은 **StatusResponseType** 이다.

<ManageNameIDResponse> 메시지는 서명되거나 또는 그렇지 않다면 메시지를 전달하는데 사용되는 프로토콜 바인딩에 의해 인증되고 무결성을 보장받아야 한다.

다음 스키마 조각은 <ArtifactResponse> 요소와 **ArtifactResponseType** 복합 타입을 정의한다.

```
<element name="ManageNameIDResponse" type="samlp:StatusResponseType"/>
```

3.6.3. 처리 규칙들

만약 요청이 수신자가 인식하지 못하는 <saml:NameID> (또는 암호화 버전)을 포함한다면, 응답 제공자는 예러 <Status>를 가지고 응답해야만 하고 urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal 의 차상위-수준 <StatusCode>를 가지고 응답할 수 있다.

만약 <Terminate> 요소가 요청에 포함되면, 요청 제공자는 (그것이 서비스 제공자인 경우) 그것이 더 이상 IdP 로부터 주장을 받아들이지 않을 것이라는 사실을 가리키거나 또는 (그것이 IdP 인 경우) 그것이 더 이상 서비스 제공자에게 사용자에게 대한 주장들을 발급하지 않을 것이라는 사실을 가리킨다. 수신하는 제공자는 이름 식별자로 나타나는 관계(relationship)가 종료되었다는 사실을 알게되고 이에 따라 해당되는 유지보수 작업을 수행할 수 있다. 수신하는 제공자는 관계가 종료된 사용자들의 활성(active) 세션(들)을 무효화시키기로 결정할 수 있다.

만약 서비스 제공자가 <NewID> (또는 <NewEncryptedID>) 요소를 포함함으로써 사용자에게 대한 서비스 제공자의 식별자가 변화될 것을 요청하면, IdP 는 사용자에게 대하여 서비스 제공자와 다음에(subsequently) 통신할 때, 요소의 내용을 SPProvidedID 로 포함해야만 한다.

만약 IdP 가 <NewID> (또는 <NewEncryptedID>) 요소를 포함함으로써 사용자에게 대한 IdP 의 식별자가 변화될 것을 요청하면, 서비스 제공자는 사용자에게 대하여 해당 IdP 와 다음에(subsequently) 통신할 때 요소의 내용을 <saml:NameID>로 포함해야만 한다.

<EncryptedID>와 <NewEncryptedID> 요소들을 사용하여, 원 식별자와 새로운 식별자는 둘다 또는 하나만 암호화될 수 있으며, 또는 하나도 암호화되지 않을 수 있다.

어떠한 경우라도, 요청에 있는 <saml:NameID> 내용과 그것과 관련된 SPProvidedID 속성은 사용자에게 대하여 제공자들 사이에 설정된 가장 최근의 이름 식별자를 포함해야만 한다.

urn:oasis:names:tc:SAML:2.0:nameidformat:persistent 를 Format 속성 값으로 가지는 식별자의 경우, NameQualifier 속성은 식별자를 생성한 IdP 의 유일한 식별자를 포함해야만 한다. 만약 이 식별자가 IdP 와 서비스 제공자가 멤버로 속한 제휴그룹(affiliation group) 사이에서 설정되었으면, SPNameQualifier 속성은 제휴그룹의 유일한 식별자를 포함해야만 한다. 만약 그렇지 않다면, SPNameQualifier 속성은 반드시 서비스 제공자의 유일한 식별자를 포함해야만 한다. 만약 이러한 속성들이 포함하는 프로토콜 메시지의 <Issuer> 요소의 값과 다르게 매치가 되면, 이러한 값들을 생략될 수 있지만, 혼동을 가져올 수 있기 때문에 이렇게 하지 않을 것이 권고된다 (NOT RECOMMENDED).

이러한 식별자들에 대한 변화가 요청자와 응답자 둘 모두의 시스템들로 전파되기까지는 상당히 많은 시간이 소요될 수 있다. 구현물들은 이름 식별자 변화가 완료될 때까지 얼마 동안 각각의 기관이 현재와 과거의 이름 식별자 모두를 받아들이는 것을 허용하기를 원할 수 있다. 이렇게 하지 않으면 사용자가 자원들을 접근하지 못하는 상황을 초래할 수 있다.

기반 요청과 응답 메시지들과 관련된 모든 다른 처리 규칙들은 준수되어야만 한다

3.7. 단일 로그아웃 프로토콜(Single Logout Protocol)

단일 로그아웃 프로토콜은 특정한 세션 기관에 의해 제공된 모든 세션들이 거의 동시에 종료될 수 있도록 해 주는 메시지 교환 프로토콜을 제공한다. 단일 로그아웃 프로토콜은 사용자가 세션 참여자에서 로그아웃을 하거나 또는 사용자가 세션 기관에서 직접 로그아웃을 할 때 사용된다. 이 프로토콜은 또한 타임아웃(timeout) 때문에 사용자를 로그아웃 시키는데도 사용될 수 있다. 로그아웃 이벤트(event)에 대한 이유는 Reason 속성을 통해 표시할 수 있다.

세션 기관에서 제공하는 인증 문장들을 포함하는 주장들을 근거로 하여, 사용자는 세션기관과 개별적인 세션 참여자들과 인증된 세션을 설정할 수 있다.

사용자가 세션 참여자에서 단일 로그아웃 절차를 호출(involve)할 때, 세션 참여자는 자신이 사용자에게 대하여 관리하는 세션과 관련된 인증 문장을 포함하는 주장을 제공한 세션 기관에게 <LogoutRequest> 메시지를 보내야만 한다

사용자가 세션 기관에서 로그아웃을 호출(involve)할 때, 또는 세션 참여자가 세션 기관에게 로그아웃 요청을 보낼 때, <LogoutRequest> 메시지를 세션 기관에게 전달했던 세션 참여자를 제외하고, 세션 기관은 자신이 현재 사용자 세션하에서 사용자에게 대한 인증 문장들을 포함하는 주장들을 제공했던 세션 참여자들 각각에게 <LogoutRequest> 메시지를 보내야 한다. 세션 기관은 이 프로토콜을 이용하여 가능하면 많은 세션 참여자에게 요청 메시지를 전달하려 해야 하고, 사용자와 관련된 자신의 세션을 종료시키고 마지막으로, 만약 있다면, 요청하는 세션 참여자에게 <LogoutResponse> 메시지를 반환해야 한다.

3.7.1. <LogoutRequest> 요소

세션 참여자 또는 세션 기관은 세션이 종료되었음을 표시하기 위해 <LogoutRequest> 메시지를 전달한다.

<LogoutRequest> 메시지는 서명되거나 또는 그렇지 않다면 메시지를 전달하는데 사용되는 프로토콜 바인딩에 의해 인증되고 무결성을 보장받아야 한다.

이 메시지는 LogoutRequestType 복합 타입을 가지며, 이것은 RequestAbstractType 을 확장하며 다음 요소들을 추가한다:

NotOnOrAfter [Optional]

요청이 종료되는 시각, 이 시각 이후에는 수신자가 이 메시지를 폐기할 수 있다. 시각 값은 1.6.3 절에서 설명된 것처럼 UTC 형식으로 인코딩된다.

Reason [Optional]

로그아웃에 대한 이유를 표시함. URI 참조 형태를 띤다.

<saml:BaseID> 또는 <saml:NameID> 또는 <saml:EncryptedID> [Required]

이 요청을 하기 이전에 IdP 와 서비스 제공자들에 의해 현재 인지되고 있는 대로 사용자를 명기하는 (평문 또는 암호화 형태인) 식별자와 관련된 속성들. (이 요소에 대한 더 많은 정보는 2.2 절을 참조)

<SessionIndex> [Optional]

메시지 수신자 측에서 이 세션을 색인하는 식별자

다음 스키마 조각은 <LogoutRequest> 요소와 관련된 LogoutRequestType 복합 타입을 정의한다.


```

<element name="LogoutRequest" type="saml:LogoutRequestType"/>

<complexType name="LogoutRequestType">
  <complexContent>
    <extension base="saml:RequestAbstractType">
      <sequence>
        <choice>
          <element ref="saml:BaseID"/>
          <element ref="saml:NameID"/>
          <element ref="saml:EncryptedID"/>
        </choice>
        <element ref="saml:SessionIndex" minOccurs="0"
          maxOccurs="unbounded"/>
      </sequence>
      <attribute name="Reason" type="string" use="optional"/>
      <attribute name="NotOnOrAfter" type="dateTime" use="optional"/>
    </extension>
  </complexContent>
</complexType>

<element name="SessionIndex" type="string"/>

```

3.7.2. <LogoutResponse> 요소

<LogoutRequest> 메시지의 수신자는 <LogoutResponse> 메시지를 가지고 응답해야만 한다. 이 요소는 추가적인 내용을 가지고 있지 않은 **StatusResponseType** 이다.

<LogoutResponse> 메시지는 서명되거나 또는 그렇지 않다면 메시지를 전달하는데 사용되는 프로토콜 바인딩에 의해 인증되고 무결성을 보장받아야 한다.

다음 스키마 조각은 <LogoutResponse> 요소와 **LogoutResponseType** 복합 타입을 정의한다.

```

<element name="LogoutResponse" type="saml:StatusResponseType"/>

```

3.7.3. 처리 규칙들

메시지 송신자는 <LogoutRequest>을 송신한 이유를 표시하기 위해 Reason 속성을 사용할 수 있다. 다음 값들은 모든 메시지 송신자에 의해 사용될 목적으로 이 표준에서 정의된다.

urn:oasis:names:tc:SAML:2.0:logout:user

사용자가 표시된 세션을 종료하기를 원하기 때문에 이 메시지가 보내진다는 것을 명기한다.

urn:oasis:names:tc:SAML:2.0:logout:admin

관리자가 사용자에게 표시된 세션을 종료시키는 것을 원하기 때문에, 이 메시지가 보내진다는 것을 명기한다.

기반 요청과 응답 메시지들과 관련된 모든 다른 처리 규칙들은 준수되어야만 한다. 추가적인 처리 규칙들은 다음 절에서 제공된다.

3.7.3.1. 세션 참여자 규칙들 (Session Participant Rules)

세션 참여자가 <LogoutRequest> 메시지를 수신하면, 세션 참여자는 메시지를 반드시 인증해야만 한다. 만약 송신자가 사용자의 현재 세션과 연결된 인증 문장을 포함하는 주장을 제공한 기관이라면, 세션 참여자는 <saml:BaseID>, <saml:NameID>, 또는 <saml:EncryptedID> 요소와 요청 메시지 내에서 제공된 어떠한 <SessionIndex> 요소들에 의해 참조되는 사용자의 세션(들)을 무효화해야만 한다. 만약 <SessionIndex> 요소가 제공되지 않는다면, 사용자와 관련된 모든 세션들은 무효화되어야만 한다.

비록 주장이 로그아웃 요청 이후에 도착할지라도, 세션 참여자는 다음 조건을 충족하는 어떠한 주장들에게도 로그아웃 요청 메시지를 적용해야만 한다.

- 3.3.4 절에서 설명한 것 처럼, 주장 주체가 <LogoutRequest>에 있는 <saml:BaseID>, <saml:NameID> 또는 <saml:EncryptedID> 요소와 강하게 매치된다(match).
- 주장의 인증 문장들 중에 하나의 SessionIndex 속성이 로그아웃 요청에서 명기된 <SessionIndex> 요소들 중에 하나와 매치되거나 또는 로그아웃 요청이 <SessionIndex> 요소를 포함하지 않는다.
- 그렇지 않다면, 특히 조건들이나 주체 확인 데이터에 있는 명기되어 있는 NotOnOrAfter 속성들의 값과 같이, 주장 자체에서 명기된 시각 조건들을 근거로하여 주장은 유효할 수 있을 것이다.
- 로그아웃 요청이 아직 만료되지 않았다 (메시지 상의 NotOnOrAfter 속성을 검사하여 판단된다).

주의: 이 규칙은 세션 참여자가 로그아웃 요청이 목표하는(target) 실질적인 - 그리고 아직도 유효할 가능성이 있는 - 주장(들)을 수신하기 전에 (<SessionIndex> 요소(들)에 의해 식별되는 대로) 세션 참여자가 단일 또는 다중 주장(들)을 목표하는(target) 로그아웃 요청을 수신하는 상황을 방지하는 것을 목적으로 한다. 로그아웃 요청 자체가 폐기될 때까지 (요청에 있는 NotOnOrAfter 값이 초과되었다) 또는 로그아웃 요청이 목표하는 주장이 수신되어 적절히 처리될 때까지, 세션 참여자는 로그아웃 요청을 존중해야 한다.

3.7.3.2. 세션 기관 규칙들 (Session Authority Rules)

세션 기관이 <LogoutRequest> 메시지를 수신하면, 세션 기관은 송신자를 반드시 인증해야만 한다. 만약 송신자가 세션 기관이 현재의 세션에 대한 인증 문장을 포함하는 주장을 제공했던 세션 참여자이면, 세션 기관은 명기된 순서대로 다음과 같이 처리를 해야 한다.

- 세션 기관이 사용자의 인증을 대리한 모든 세션 기관에게 <LogoutRequest> 메시지를 전달한다. 단 <LogoutRequest>를 세션 기관에서 전달한 개시자(originator)는 전달 대상에서 제외된다.
- 현재 <LogoutRequest> 메시지를 세션 기관에게 전달한 세션 참여자를 제외하고, 세션 기관이 현재 세션에서 주장들을 제공한 세션 참여자 각각에게 <LogoutRequest> 메시지를 전달한다.
- <saml:BaseID>, <saml:NameID>, 또는 <saml:EncryptedID> 요소와 요청 메시지 내에서 제공된 어떠한 <SessionIndex> 요소들에 의해 참조되는 사용자의 세션(들)을 종료시킨다.

만약 세션 기관이 그 자신에 대한 사용자의 세션을 성공적으로 종료시키면, 세션 기관은 urn:oasis:names:tc:SAML:2.0:status:Success 를 값으로 가지는 최상위 수준의 상태 코드를 포함하는 <LogoutResponse> 메시지를 가지고, 만약 있다면, 원래(original) 요청자에게 응답해야만 한다. 만약 세션 기관이 그렇게 할 수 없다면, 그것은 에러를 표시하는 최상위-수준 상태 코드를 포함하는 <LogoutResponse> 메시지를 가지고 응답해야만 한다. 이와 같이, 최상위-수준 상태는 단지 세션 기관 자체에 관한 로그아웃 연산의 상태를 가리킨다.

비록 이러한 시도들 중에 하나 또는 그 이상이 실패하거나 또는 (예를 들어, 원래 요청이 모든 참여자들에게 로그아웃이 전파될 수 없도록 하는 프로토콜 바인딩을 사용하는 것이기 때문에) 시도될 수 없다고 할지라도, 세션 기관은 적용가능한/사용할 수 있는 어떠한 프로토콜 바인딩을 사용하여서라도 각각의 세션 참여자와 접촉하려 시도해야 한다.

모든 세션 참여자가 이러한 <LogoutRequest> 메시지들에 응답하지는 못할 경우에 (또는 만약 모든 참여자들이 접촉이 될 수 없다면), 세션 기관은 모든 다른 세션 참여자들이 로그아웃의 확인으로 응답하는데 성공하지는 못했다는 것을 가리키기 위해 urn:oasis:names:tc:SAML:2.0:status:PartialLogout 을 값으로 갖는 차상위-수준 상태 코드를 그것의 <LogoutResponse> 메시지에 포함해야만 한다.

세션 기관은 세션 참여자로부터 <LogoutRequest>를 수신받지도 않고 다른 이유들 때문에 로그아웃을 시작(initiate)할 수 있다. 다음과 같은 것들이 로그아웃 시작 이유들에 포함된다.

- 만약 얼마간의 타임아웃 기간(timeout period)이 어떤 개별적인 세션 참여자와 대역외 방식(out-of-band)으로 협의가 되었다면, 세션 기관은 해당되는 참여자에게만 <LogoutRequest>를 전달할 수 있다.
- 협의된 전역 타임아웃 기간이 초과되었다.
- 사용자 또는 어떤 다른 신뢰되는 엔티티가 세션 기관에서 사용자의 로그아웃을 직접적으로 요청했다.
- 세션 기관이 사용자의 크리덴셜들이 누설되었다(compromised) 다고 판단했다.

로그아웃 요청 메시지를 구성할 때, 세션 기관은 메시지에 대한 만료시각을 가리키도록, 메시지의 NoTOOnOrAfter 의 값에 시각 값을 설정해야만 한다. 이렇게 함으로써, 이 시각 이후에는 로그아웃 요청이 수신자에 의해 폐기될 수 있다. 이 값은 (로그아웃 요청의 SessionIndex 속성에서 지시된 것처럼) 목표하는 세션의 일부로써 가장 최근에 발급된 주장에서 명기된 어떠한 NotOnOrAfter 속성의 값보다도 더 크거나 또는 동일한 값으로 설정해야 한다.

Reason 속성으로 3.6.3 절에서 명기된 값 이외에, 다음 값들이 또한 세션 기관에 의해서만 사용되는 것이 가능하다.

urn:oasis:names:tc:SAML:2.0:logout:global-timeout

전역 세션 타임아웃 기간이 초과되었기 때문에 이 메시지가 전송되었다는 것을 명기한다.

urn:oasis:names:tc:SAML:2.0:logout:sp-timeout

참여자와 세션 기관 사이에 협의된 타임아웃 기간이 초과되었기 때문에, 이 메시지가 전송되었다는 것을 명기한다.

3.8. 이름 식별자 매핑 프로토콜(Name Identifier Mapping Protocol)

사용자에 대한 식별자를 IdP 와 공유하는 어떤 엔티티가 특정 형태 또는 연계 네임스페이스(federation namespace)로 동일한 사용자에 대한 이름 식별자를 얻고자 할 때, 이 엔티티는 이 프로토콜을 사용하여 IdP 에게 요청을 전달할 수 있다.

예를 들어, 사용자에 대한 식별자를 공유하지 않는 다른 서비스 제공자와 통신하기를 원하는 서비스 제공자는 두 서비스 제공자 모두와 사용자에 대한 식별자를 공유하는 IdP 를 이용하여 자신이 소유한 사용자 식별자로부터 새로운 식별자를 매핑할 수 있다. 일반적으로 새로운 식별자는 암호화되며, 이것을 이용하여 서비스 제공자는 다른 서비스 제공자와 통신할 수 있다.

관련된 식별자 타입에 관계없이, 특정한 배치가 암호화와 같은 보호가 필요 없다고 명시하지 않으면, 매핑된 식별자는 <saml:EncryptedID> 요소로 암호화되어야 한다.

3.8.1. <NameIDMappingRequest> 요소

IdP로부터 사용자에 대한 다른 이름 식별자를 요청하기 위해, 요청자는 <NameIDMappingRequest> 메시지를 전송한다. 이 메시지는 **NameIDMappingRequestType** 복합 타입을 가지며, 이것은 **RequestAbstractType**을 확장하며 다음 요소들을 추가한다:

<saml:BaseID> 또는 <saml:NameID> 또는 <saml:EncryptedID> [Required]

이 요청을 하기 이전에 IdP와 서비스 제공자들에 의해 현재 인지되고 있는 대로 사용자를 명기하는 (평문 또는 암호화 형태인) 식별자와 관련된 설명 데이터. (이 요소에 대한 더 많은 정보는 2.2 절을 참조)

<NameIDPolicy> [Required]

반환되는 이름 식별자에 대한 포맷과 선택적인 항목인 이름 한정자에 대한 요구사항들
다음 스키마 조각은 <NameIDMappingRequest> 요소와 **NameIDMappingRequestType** 복합 타입을 정의한다.

```
<element name="NameIDMappingRequest"
  type="samlp:NameIDMappingRequestType"/>

<complexType name="NameIDMappingRequestType">
  <complexContent>
    <extension base="samlp:RequestAbstractType">
      <sequence>
        <choice>
          <element ref="saml:BaseID"/>
          <element ref="saml:NameID"/>
          <element ref="saml:EncryptedID"/>
        </choice>
        <element ref="samlp:NameIDPolicy"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

3.8.2. <NameIDMappingResponse> 요소

<NameIDMappingRequest> 메시지의 수신자는 <NameIDMappingResponse> 메시지를 가지고 응답해야만 한다. 이 요소는 **NameIDMappingResponseType** 복합타입이며, 이것은 **StatusResponseType** 을 확장하고 다음 요소를 추가한다:

<saml:NameID> 또는 <saml:EncryptedID> [Required]

요청된 방식으로 사용자를 명기하는 식별자 및 관련된 속성들. 일반적으로 암호화된 형식임. (이 요소에 대한 더 많은 정보는 2.2 절을 참조)

이 메시지는 서명되거나 또는 그렇지 않다면 메시지를 전달하는데 사용되는 프로토콜 바인딩에 의해 인증되고 무결성을 보장받아야 한다.

다음 스키마 조각은 <NameIDMappingResponse> 요소와 **NameIDMappingResponseType** 복합 타입을 정의한다.

```
<element name="NameIDMappingResponse"
  type="samlp:NameIDMappingResponseType"/>

<complexType name="NameIDMappingResponseType">
  <complexContent>
    <extension base="samlp:StatusResponseType">
      <choice>
        <element ref="saml:NameID"/>
        <element ref="saml:EncryptedID"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

3.8.3. 처리 규칙들

만약 응답자가 요청에서 식별되는 사용자를 인식하지 못하면, 응답자는 urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal 을 값으로 갖는 차상위-수준 <StatusCode>를 포함하는 예러 <Status>를 가지고 응답할 수 있다.

응답자의 판단에 따라, urn:oasis:names:tc:SAML:2.0:status:InvalidNameIDPolicy 상태 코드가 요청되는 포맷 또는 네임스페이스로 식별자를 제공할 수 없거나 또는 하지 않을 것임을 표시하기 위해 반환될 수 있다.

기반 요청과 응답 메시지들과 관련된 모든 다른 처리 규칙들은 준수되어야만 한다

4. SAML 버전

SAML 표준 집합은 두 가지 독립적인 방식으로 버전이 부여된다. 버전 차이들 발견하고 처리하는 처리 규칙들과 함께, 각각의 방식에 대하여 이번 절에서 논의된다. 또한 표준에 대한 미래의 개정들에서 언제 그리고 왜 특정 버전 정보가 변화될 것으로 예상되는지에 대한 가이드라인들도 또한 포함된다.

버전 정보가 주(Major) 버전과 부(Minor) 버전 두 가지 모두를 사용하여 표현될 때, 버전 정보는 Major.Minor 형태로 표현된다. MajorB.MinorB 버전 번호가 MajorA.MinorA 버전 번호 보다 더 높을 수 있는 필요충분 조건은 다음과 같다:

(MajorB > MajorA) 또는 ((MajorB = MajorA) 그리고 (MinorB > MinorA))

4.1. SAML 표준 집합 버전(SAML Specification Set Version)

SAML 표준의 각각의 배포(release)는 SAML 표준의 이전 또는 이후 버전들과의 관계를 설명하는 주 그리고 부 버전 지정(major and minor version designation)을 포함할 것이다. 이 버전은 이 표준의 내용에 표현될 것이다. 이 표준안에 대한 변경의 전체 크기와 범위는 일단의 변화가 주 또는 부 개정으로 구성되는지 여부를 비공식적으로 알릴 것이다. 일반적으로, 만약 표준이 이전 표준 집합과 후방(backward) 호환이 된다면, 그러면 새로운 버전은 부 개정일 것이다. 만약 그렇지 않다면, 변화들은 주 개정을 구성할 것이다.

이 표준안은 버전 V2.0 을 정의한다.

4.1.1. 스키마 버전(Schema Version)

비-규범적인 메커니즘으로써, 표준 집합의 일부로써 발행된 어떠한 XML 스키마 문서들로 Major.Minor 형태의 값을 가지는 <xs:schema> 요소의 version 속성을 포함할 것이며, 이것은 스키마 문서가 발행된 표준 집합의 버전을 반영한다. 검증하는 구현물들은 어떠한 버전의 스키마가 메시지를 검증하는데 사용되는지 구별하는 수단으로 이 속성을 사용하거나 또는 동일한 논리 스키마의 다중 버전들을 지원하기 위해 이 속성을 사용할 수 있다.

4.1.2. SAML 주장 버전(SAML Assertion Version)

SAML <Assertion> 요소는 Major.Minor 형태의 문자열로 주장의 주/부 버전을 표현하기 위한 속성을 가지고 있다. SAML 표준 집합의 각각의 버전은 동일한 버전의 주장들에 대한 문법, 의미 그리고 처리 규칙들을 문서화하기 위하여 해석될 수 있다. 즉, 표준 집합 버전 1.0 은 주장 버전 1.0 과 기타 등등을 설명한다.

주장 버전과 그 주장 버전에 대한 스키마 정의들을 위해 명기된 목표 XML 네임스페이스 사이에는 명시적으로 어떠한 관계도 없다.

다음 처리 규칙이 적용된다:

- SAML 보장하는 기관은 그 기관에서 지원하지 않는 전체 Major.Minor 주장 버전 숫자(number)를 가지는 어떠한 주장도 발급한지 알아야만 한다.
- SAML 의지하는 기관은 의지하는 기관이 지원하지 않는 주 주장 버전 번호를 가지는 어떠한 주장도 처리하지 않아야만 한다.
- SAML 의지하는 기관은 의지하는 기관이 지원하는 부 주장 버전 숫자보다 더 높은 부 주장 버전 숫자를 가지는 주장을 처리할 수도 있고 거절할 수 있다. 하지만, 주 주장 버전 숫자를 공유하는 모든 주장들은 동일한 일반 처리 규칙들과 의미들을 공유해야만 하고 구현에 의해 동일한 형태로 취급될 수 있다. 예를 들어, 만약 V1.1 주장이 V1.0 주장의 문법을 공유한다면, 구현은 악 영향(ill effect)을 발생시키지 않으면서도 그 주장을 V1.0 주장으로 처리할 수 있다.

4.1.3. SAML 프로토콜 버전(SAML Protocol Version)

다양한 SAML 프로토콜들의 요청과 응답 요소들은 Major.Minor 형태의 문자열을 사용하여 요청 또는 응답 메시지의 주 그리고 부 버전을 표시하기 위한 속성을 포함한다. SAML 표준 집합의 각각의 버전은 동일한 버전의 주장들에 대한 문법, 의미 그리고 처리 규칙들을 문서화하기 위하여 해석될 수 있다. 즉, 표준 집합 버전 1.0 은 주장 버전 1.0 과 기타 등등을 설명한다.

프로토콜 버전과 그 프로토콜 버전에 대한 스키마 정의들을 위해 기술된 목표 XML 네임스페이스 사이에는 명시적으로 어떠한 관계도 없다.

SAML 프로토콜 요청과 응답 요소들에서 사용된 버전 숫자는 SAML 표준 집합의 어떠한 특정 개정과 매칭될 것이다.

4.1.3.1. 요청 버전(Request Version)

다음 처리 규칙들은 요청들에 적용된다:

- SAML 요청자는 SAML 요청자와 SAML 응답자 둘 모두에 의해 지원되는 가장 높은 버전을 가지는 요청들을 발급해야 한다.
- 만약 SAML 요청자가 SAML 응답자의 능력들을 알지 못하면, SAML 요청자는 응답자가 자신에 의해 지원하는 가장 높은 버전을 가지는 요청들을 지원할 것으로 가정해야 한다.

SAML 응답자는 그것이 지원하는 가장 높은 요청 버전보다 부 버전 숫자가 더 높은 요청은 처리할 수 있고 또는 거절할 수 있다. 하지만, 주 요청 버전 숫자를 공유하는 모든 요청들은 동일한 일반 처리 규칙들과 의미들을 공유해야만 하고 구현에 의해 동일한 형태로 취급될 수 있다. 예를 들어, 만약 V1.1 요청이 V1.0 요청의 문법을 공유한다면, 응답자는 악 영향(ill effect)을 발생시키지 않으면서도 그 요청 메시지를 V1.0 요청 메시지로 처리할 수 있다.

4.1.3.2. 응답 버전(Response Version)

다음 처리 규칙들은 응답들에 적용된다:

- SAML 응답자는 대응되는 요청 메시지의 요청 버전 숫자보다 더 높은 버전 숫자를 가지는 응답으로 응답 메시지를 발급하지 않아야만 한다.
- SAML 응답자는 urn:oasis:names:tc:SAML:2.0:status:RequestVersionTooHigh 에러를 보고하는 것 외에, 대응되는 요청 메시지의 주 요청 버전 숫자보다 낮은 주 응답 버전 숫자를 가지는 응답 메시지를 발급하지 않아야만 한다.
- 호환되지 않는 SAML 프로토콜 버전들로부터 유발되는 에러 응답은 urn:oasis:names:tc:SAML:2.0:status:VersionMismatch 를 최상위-수준 <StatusCode> 값으로 보고해야만 하고 다음과 같은 차상위-수준 값들 중에 하나를 보고할 수 있다:

urn:oasis:names:tc:SAML:2.0:status:RequestVersionTooHigh,

urn:oasis:names:tc:SAML:2.0:status:RequestVersionTooLow, 또는

urn:oasis:names:tc:SAML:2.0:status:RequestVersionDeprecated.

4.1.3.3. 허용가능한 버전 조합들(Permissible Version Combinations)

SAML 주장 네임스페이스가 SAML 프로토콜 스키마에 임포트(import)되는 것이 허가하는 것처럼, 특정 주 버전을 가지는 주장들은 동일한 주 버전을 가지는 응답 메시지들에서만 나타난다. 예를 들어, 만약 적절한 주장 스키마가 네임스페이스 임포트중에 참조된다면, V1.1 주장은 V1.0 응답 메시지에 포함될 수 있고 V1.1 응답 메시지에 V1.0 주장이 있을 수도 있다. 그러나, V1.0 주장은 V2.0 응답 메시지에서는 나타나지 않아야만 한다. 왜냐하면 이들이 서로 다른 주 버전을 가지고 있기 때문이다.

4.2. SAML 네임스페이스 버전(SAML Namespace Version)

표준 집합의 일부로써 발행된 XML 스키마 문서들은 타입, 요소, 속성 정의들은 그것 안에 위치하게 되는 하나 또는 그 이상의 목표 네임스페이스들을 포함한다. 각각의 네임스페이스는 다른 네임스페이스들과 구별되고, 표준의 일부를 구성하는 구조적인 그리고 문법적인 정의들을 간략히 나타낸다.

표준이 정의하는 네임스페이스 URI 참조들은 일반적으로 그 URI 어디엔가 Major.Minor 형태의 버전 정보를 포함할 것이다. URI 에 있는 주 버전과 부 버전은 네임스페이스가 처음 도입되고 정의된 표준 스펙의 주 그리고 부 버전에 대응되어야만 한다. 이 정보는 전형적으로 XML 프로세스에 의해 소비되지 않고, XML 프로세서는 네임스페이스를 불투명하게 처리할 것이다. 그러나 이 정보는 표준 집합과 그것이 정의한 네임스페이스 간의 관계를 교환하는 것을 의도하고 있다. 또한 8 장에서 나열된 SAML 에서 정의된 URI-기반 식별자들이 이 패턴을 따른다.

일반적으로, 구현자들은 표준의 주 개정을 통해 정의된 네임스페이스들과 관련된 스키마 정의가 그 표준의 부 개정들에서 유효하고 안정될 것으로 기대할 수 있다. 새로운 네임스페이스들은 도입될 수 있고, 필요할 때, 이전 네임스페이스들을 대체할 수도 있지만 이러한 것은 거의 기대되지 않는다. 그러한 경우에, 이전 네임스페이스들과 그들과 관련된 정의들은 주 표준 집합 개정 때까지는 유효한 것으로 남아있는 것이 기대되어야 한다.

일반적으로, 스키마의 내용을 추가하거나 또는 변경하면서도 네임스페이스 안정성을 유지하는 것은 상호 배타적인 목표가된다. 어떤 디자인 전략들은 이와 같은 변화를 촉진시키면서, 얼마나 오래된 구현들이 주어진 변화와 반응할지를 예측하는 것을 어렵게된다. 이것은 순방향 상호운용성을 달성하는 것을 어렵게 한다. 그럼에도 불구하고, 부 개정에서 이와 같은 변화를 만들 수 있는 권한은 네임스페이스 안정성 측면에서 보존(reserved)된다. 예를 들어, 주요한 결함이나 또는 에러를 고치는 것과 같은 특수한 환경을 제외하고는, 구현들은 부 개정에서 순방향 호환적인 스키마 변화를 예측해야 하며, 새로운 메시지가 이전 스키마에 대하여 유효하도록 허가해야 한다.

구현들은 새로운 확장이나 메시지 타입에 놓여진 처리 규칙에 부합하면서 이들을 처리할 수 있다는 것을 예측하고 준비해야 한다. 부 개정은 이 표준에서 서명되는 확장 기능들을 확용하는 새로운 타입을 도입할 수 있다. 이전 구현은 그들이 강제적인 의미를 가하는 문맥에서 이들 확장을 처리해야 하는 경우에는 자연스럽게 이와 같은 확장을 거부해야 한다. 새로운 질의, 문장 또는 조건 타입들이 이들 예에 포함된다.

5. SAML과 XML 서명 문법 및 처리

SAML 주장들과 SAML 프로토콜 요청과 응답 메시지들은 서명될 수 있으며, 이 경우 다음과 같은 이점들을 가질 수 있다. 보장하는 기관에 의해 서명된 주장은 주장 무결성과 SAML 의지하는 기관에게 보장하는 기관의 인증을 제공하고, 만약 서명이 SAML 기관의 공개키-개인키 쌍을 통해 이루어지면, 출처(origin)에 대한 부인봉쇄(non-repudiation)를 제공한다. 메시지 개시자(originator)에 의해 서명된 SAML 프로토콜 요청 또는 응답 메시지는 메시지 무결성과 목적지에게 메시지 출처의 인증을 제공하고, 만약 서명이 개시자(originator)의 공개키-개인키 쌍에 의지한다면, 출처(origin)에 대한 부인봉쇄(non-repudiation)를 제공한다.

SAML 에서 항상 전자서명이 요구되는 것은 아니다. 예를 들어, 서명되지 않은 주장이 그것을 포함하는 프로토콜 메시지에 대한 서명을 통해 보호받을 수 있는 때와 같은 그러한 일부 환경에서는, 서명이 금지될 수도 있다. (주장과 같이) 포함된 개체가 비-일시적인 수명(non-transitory lifetime)을 가지도록 의도되었을 때에는, 금지된 서명들을 사용하는데 주의를 기울여야 한다. 그 이유는 전체 문맥이 검증을 위해 보유되어야만 하기 때문이며, 이렇게 하는 것은 XML 내용을 노출시키고 잠재적으로 불필요한 오버헤드를 추가시키게 된다. 또 다른 예로써, SAML 의지하는 기관 또는 SAML 요청자는 SAML 보장하는 기관 또는 SAML 응답자로부터 직접적으로 (어떠한 중개자도 없이) 안전한 채널을 통해 주장과 프로토콜 메시지를 얻었을 지도 모르며, 이 경우 보장하는 기관 또는 SAML 응답자가 전자서명이 아닌 다른 수단을 통해 의지하는 기관 또는 SAML 응답자에게 인증한다.

많은 다른 기술들이 두 기관 사이에 직접적인 인증과 안전한 채널을 설립하는데 이용가능하다. 이런 기술에는 TLS, HMAC, 패스워드-기반 메커니즘 등을 포함한다. 또한, 적용가능한 보안 요구사항들은 통신 응용과, 주장 또는 전달된 메시지의 성질(nature)에 따라 좌우된다. 모든 다른 문맥에서, 전자 서명들은 주장과 요청/응답 메시지들에 대하여 사용되는 것이 권고된다. 특히:

- SAML 보장하는 기관이 아닌 엔티티로부터 SAML 의지하는 기관이 얻은 SAML 주장은 SAML 보장하는 기관에 의해 서명되어야 한다.
- 원래의(originating) 송신자가 아닌 엔티티로부터 목적지에 도착한 SAML 프로토콜 메시지는 송신자에 의해 서명되어야 한다.
- 프로파일들은 SAML 문서들을 포함하는 S/MIME 또는 서명된 자바 객체들과 같은 다른 서명 메커니즘들을 명기할 수 있다. 문맥과 상호운용성을 보유하는 것에 대한 경고(caveats)가 적용된다. XML 서명들은 주요한 SAML 서명 메커니즘이 되도록 의도되었지만, 이 표준은 다른 메커니즘들을 요구할 지도 모르는 프로파일들과의 상호운용성을 보장하도록 시도한다.
- 프로파일이 다른 서명 메커니즘을 명기하지 않는다면, XML 전자 서명이 포장되어(enveloped)야만 한다.

5.1. 주장 서명하기 (Signing Assertions)

모든 SAML 주장들은 XML 서명을 이용하여 서명될 수 있다. 이것은 1 장에서 설명된 것처럼, 주장 스키마에 반영된다.

5.2. 요청/응답 서명하기

모든 SAML 프로토콜 요청과 응답 메시지들은 XML 서명을 사용하여 서명될 수 있다.

5.3. 서명 상속(Signature Inheritance)

SAML 주장은 다른 SAML 요소들 내에 내장(embedded)될 수 있다. 감싸는(enclosing) <Assertion> 또는, 요청 또는 응답과 같이 SAML 주장을 내장하는 다른 SAML 요소들 역시 서명될 수 있다. SAML 주장이 <ds:Signature> 요소를 포함하지 않지만 자신을 포함하는(enclosing) SAML 요소가 <ds:Signature> 요소를 포함하고, 서명이 <Assertion> 요소와 그것의 모든 자식들에게 적용이 되면, 주장은 자신을 포함하는 요소로부터 서명을 상속받은 것으로 간주될 수 있다. 이 경우, 주장 자체가 동일한 키와 서명 선택들(options)로 서명된 것으로 해석되어야 한다.

많은 SAML 사용 케이스들은 서명된 SOAP 메시지들, S/MIME 패키지들 그리고 인증된 TLS 연결들과 같은 다른 보호된 데이터 구조들 내에 포함된(enclosed) SAML XML 데이터를 수반한다. SAML 프로파일들은 서명 상속이나 또는 주위 문맥으로부터 다른 인증 정보로써 SAML 요소들을 해석하기 위하여 추가적인 규칙들을 정의할 수 있다. 그러나, 프로파일에 의해 별다르게 식별되지 않는다면, 이와 같은 상속은 추론되어(inferred)서는 안된다.

5.4. XML 서명 프로파일(XML Signature Profile)

W3C XML Signature:2002 는 유연하고 많은 선택을 가지고 데이터를 서명하기 위한 일반적인 XML 문법을 제공한다. 이 절은 이러한 기능에 대한 제약사항들을 자세히 언급하여, SAML 프로세서가 XML 서명 처리의 일반적인 전체 기능을 다룰 필요가 없게 한다. 이 사용법은 서명들이 적용될 수 있는 루트 요소들 상에 있는 **xs:ID** 타입의 속성들을 특수하게 사용한다. 특히, <Assertion>과 다양한 요청 및 응답 요소들에 있는 ID 속성들이 해당된다. 이들 속성들은 집합적으로 이 절에서 식별자 속성들이라고 불린다.

이 프로파일은 SAML 주장, 요청들 그리고 응답들 내에서 집적적으로 발견되는 <ds:Signature> 요소들의 사용에만 적용된다. 서명이 다른 곳에서 나타지만 SAML 내용에 적용되는 그러한 다른 프로파일들에서는 자유롭게 다른 방식들이 정의될 수 있다.

5.4.1. 서명 포맷들과 알고리즘들(Signing Formats and Algorithms)

XML 서명은 서명을 문서와 연관시키는데 세 가지 방법을 사용한다: 포함하기(enveloping), 포함되기(enveloped), 그리고 분리되기(detached).

SAML 주장들과 프로토콜들은 주장들과 프로토콜 메시지들을 서명할 때, 반드시 포함되는 서명들(enveloped signatures)를 사용해야만 한다. SAML 처리기들은 <http://www.w3.org/2000/09/xmlsig#rsa-sha1> 로 식별되는 알고리즘에 따라 공개키 연산들에 대한 RSA 서명과 검증을 사용하는 것을 지원해야 한다.

5.4.2. 참조들(References)

SAML 주장들과 프로토콜 메시지들은 서명되는 주장과 프로토콜 메시지의 루트 요소에 있는 ID 속성에 대해 값을 제공해야만 한다. 주장 또는 프로토콜 메시지의 루트 요소는 서명된 주장 또는 프로토콜 메시지를 포함하는 실질적인 XML 문서의 루트 요소일 수도 있고 아닐 수도 있다 (즉, 이것이 SOAP 봉투(envelope) 내에 포함될 수 있다).

서명들은 서명되는 주장 또는 프로토콜 메시지의 루트 요소의 ID 속성 값에 대한 동일-문서 참조(same-document reference)를 포함하는 단일한 <ds:Reference>를 포함해야만 한다. 예를 들어, 만약 ID 속성 값이 “foo” 이면, 그러면 <ds:Reference> 요소의 URI 속성은 “#foo” 이어야만 한다.

5.4.3. 정규화 방법(Canonicalization Method)

SAML 구현들은 <ds:SignedInfo>의 <ds:CanonicalizationMethod> 요소 안에서 그리고 <ds:Transform> 알고리즘으로써, 두 가지 모두에서, 주석을 가지는 또는 가지지 않는 배제하는 정규화(Exclusive Canonicalization)를 사용해야 한다. 배제하는 정규화의 사용은 어떠한 XML 문맥에 내장된 SAML 메시지에 대하여 생성된 서명들이 그 문맥에 독립적으로 검증될 수 있도록 하는 것을 보장한다.

5.4.4. 변형들(Transforms)

SAML 메시지에 있는 서명들은 (<http://www.w3.org/2000/09/xmldsig#enveloped-signature> 식별자를 가진) 포함되는(enveloped) 서명 변형이나 또는 (<http://www.w3.org/2001/10/xml-exc-c14n#> 또는 <http://www.w3.org/2001/10/xml-exc-c14n#WithComments> 식별자를 가진) 배제하는(exclusive) 정규화 변형들이 아닌 다른 변형들을 포함하지 않아야 한다.

서명들의 검증자는 다른 변형 알고리즘들을 포함하는 서명들은 유효하지 않은 것으로 거절할 수 있다. 만약 검증자들이 거절하지 않으면, 그들은 SAML 메시지의 어떠한 내용도 서명으로부터 배제되지 않는다는 것을 보장해야만 한다. 이것은 어떤 변형들이 받아들여질 수 있는지에 대하여 대역외 방식(out-of-band) 협정을 설립하거나 또는 수작업으로 내용에 변형들을 적용하고 그 결과를 동일한 SAML 메시지를 구성하는 것으로써 재검증함으로써 성취될 수 있다.

5.4.5. 키정보(KeyInfo)

XML 서명은 <ds:KeyInfo> 요소의 사용법을 정의한다. SAML 은 <ds:KeyInfo>의 사용을 요구하지도 않으며 또한 그것의 사용에 대하여 어떠한 제약도 가하지 않는다. 따라서, <ds:KeyInfo>는 없을 수도 있다.

5.4.6. 예

다음은 서명된 주장을 포함하는 서명된 응답의 예이다. 줄 바꿈은 가독성을 위해 추가되었다; 즉, 서명은 유효하지 않으며 성공적으로 검증될 수 없다.


```

<Response
  IssueInstant="2003-04-17T00:46:02Z" Version="2.0"
  ID="_c7055387-af61-4fce-8b98-e2927324b306"
  xmlns="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
  <saml:Issuer>https://www.opensaml.org/IDP"</saml:Issuer>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo>
    <ds:CanonicalizationMethod
      Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
    <ds:SignatureMethod
      Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
    <ds:Reference URI="#_c7055387-af61-4fce-8b98-e2927324b306">
      <ds:Transforms>
        <ds:Transform
          Algorithm=
"http://www.w3.org/2000/09/xmldsig#envelopedsignature" />
        <ds:Transform
          Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
          <InclusiveNamespaces PrefixList="#default saml ds xs xsi"
            xmlns="http://www.w3.org/2001/10/xml-exc-
              c14n#" />
        </ds:Transform>
      </ds:Transforms>
      <ds:DigestMethod
        Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      <ds:DigestValue>TCDVSuG6grhyHbzhQFWFzGrxIPE=</ds:DigestValue>
    </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue>

```

x/GyPbzmFEe85pGD3c1aXG4VspB9V9jGCjwcRCKrtwPS6vdVNCcY5rHaFPYWkf+

5

ElYcPzx+pX1h43SmwviCqXRjRtMANWbHLhWAptaK1ywS7gFgsD01qjyen3CP+m3

D

w6vKhaqlEdl0BYrIzb4KkHO4ahNyBVXbJwqv5pUaE4=

</ds:SignatureValue>

<ds:KeyInfo>

<ds:X509Data>

<ds:X509Certificate>

MIICyJCCAJOgAwIBAgICAnUwDQYJKoZIhvcNAQEEBQAwgaxCzAJBgNVBAYTAIV

T

MRlwEAYDVQQIEwIXaXNjb25zaW4xEDAOBgNVBACTB01hZGZlb24xIDAeBgNVBA

OT

F1VuaXZlcnNpdHkgb2YgV2lzY29uc2luMSswKQYDVQQL

bmZvcm1hdGlvbiBUZWNobm9sb2d5MSUwIwYDVQQDExxIRVBLSSBTZXJ2ZXIgcQ

0Eg

LS0gMjAwMjA3MDFBMB4XDTAyMDcyNjA3Mjc1MVoXDTA2MDkwNDA3Mjc1MVow

gYsx

CzAJBgNVBAYTAIVTMREwDwYDVQQIEwhNaWNoaWdhbjESMBAGA1UEBxMJQW

5ulEFy

Ym9yMQ4wDAYDVQQKEwVWQ0FJRDEcMBoGA1UEAxMTc2hpYjEuaW50ZXJuZX

QyLmVk

dTE_nMCUGCSqGS**I**b3DQEJARYYcm9vdEBzaGliMS5pbnRlcm5ldDIuZWRR1MIGfMA

OG

CSqGS1b3DQEBAQUAA4GNADCBiQKBgQDZSAb2sxvhAXnXVIVTx8vuRay+x50z7

GJj

IHRYQglv6lqaGG04eTcyVMhoekE0b45QgvBlaOAPSZBI13R6+KYiE7x4XAWlrCP+

c2MZVeXeTgV3Yz+USLg2Y1on+Jh4HxwkPFmZBctyXiUr6DxF8rvoP9W7O27rhRjE

pmqOIfGTWQIDAQABox0wGzAMBgNVHRMBAf8EAjAAMAsGA1UdDwQEAwIFoDA

NBgkq

hkiG9w0BAQQFAAOBgQBfDqEW+OI3jqBQHIBzhujN/PizdN7s/z4D5d3pptWDJf2n

qgi7IFV6MDkhmTvTqBtjmNk3No7v/dnP6Hr7wHxvCCRWubnmIfZ6QZAv2FU78pLX

8l3bsbmRAUg4UP9hH6ABVq4KQKMknxu1xQxLhpR1ylGPdiowMNTrEG8cCx3w/w

==

</ds:X509Certificate>

</ds:X509Data>

</ds:KeyInfo>

</ds:Signature>

<Status>

<StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>

</Status>

<Assertion ID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"

IssueInstant="2003-04-17T00:46:02Z" Version="2.0"

xmlns="urn:oasis:names:tc:SAML:2.0:assertion">

<Issuer>https://www.opensaml.org/IDP</Issuer>

<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">

<ds:SignedInfo>

<ds:CanonicalizationMethod

Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">

<ds:SignatureMethod

Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>

<ds:Reference URI="#_a75adf55-01d7-40cc-929f-

dbd8372ebdfc">

```

<ds:Transforms>
  <ds:Transform
    Algorithm=

"http://www.w3.org/2000/09/xmldsig#envelopedsignature"/>
    <ds:Transform
      Algorithm=
"http://www.w3.org/2001/10/xml-exc-c14n#">
      <InclusiveNamespaces
        PrefixList="#default saml ds xs xsi"
        xmlns="http://www.w3.org/2001/10/xml-
exc-c14n#" />
      </ds:Transform>
    </ds:Transforms>
    <ds:DigestMethod
      Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <ds:DigestValue>Kclet6XcaOgOWXM4gty6/UNdvil=</ds:DigestValue>
  </ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>

hq4zk+ZknjggCQgZm7ea8fI79gJEsRy3E8LHDpYXWQlgZpkJN9CMLG8ENR4Nr+
n

7iyzixBvKXX8P53BTCT4VghPBWhFYSt9tHWu/AtJfOTh6qaAsNdeCyG86jmtP3TD
MwuL/cBUj2OtBZOQMFN7jQ9YB7kllz3RqVL+wNmeWI4=
</ds:SignatureValue>
  <ds:KeyInfo>
    <ds:X509Data>
      <ds:X509Certificate>

MIICyJCCAjOgAwIBAgICAnUwDQYJKoZIhvcNAQEEBQAwwgaxCzAJBgNVBAYTAIV
T

```

MRlwEAYDVQQIEwIXaXNjb25zaW4xEDAOBgNVBAcTB01hZGlzb24xIDAeBgNVBA
oT

F1VuaXZlcnNpdHkgb2YgV2IzY29uc2luMSswKQYDVQQLEyJEaXZpc2lvbiBvZiBJ
bmZvcm1hdGlviBUZWNobm9sb2d5MSUwIwYDVQQDExxIRVBLSSBTZXJ2ZXIlgQ
0Eg

LS0gMjAwMjA3MDFBMB4XDTAyMDcyNjA3Mjc1MVoXDTA2MDkwNDA3Mjc1MVow
gYsx

CzAJBgNVBAYTAIVTMREwDwYDVQQIEWhNaWNoaWdhbjESMBAGA1UEBxMJQW
5uIEFy

Ym9yMQ4wDAYDVQQKEwVWQ0FJRDEcMBoGA1UEAxMTc2hpYjEuaW50ZXJuZX
QyLmVk

dTEEnMCUGCSqGSib3DQEJARYYcm9vdEBzaGliMS5pbnRlcm5ldDluZWR1MIGfMA
0G

CSqGSib3DQEBAQUAA4GNADCBiQKBgQDZSAb2sxvhAXnXVIVTx8vuRay+x50z7
GJj

IHRYQglv6lqaGG04eTcyVMhoekE0b45QgvBlaOAPSZBI13R6+KYiE7x4XAWlrCP+
c2MZVeXeTgV3Yz+USLg2Y1on+Jh4HxwkPFmZBctyXiUr6DxF8rvoP9W7O27rhRjE
pmqOIfGTWQIDAQABox0wGzAMBgNVHRMBAf8EAjAAMAsGA1UdDwQEAwIFoDA
NBgkq

hkiG9w0BAQQFAAOBgQBfDqEW+OI3jqBQHIBzhujN/PizdN7s/z4D5d3pptWDJf2n

qgi7IFV6MDkhmTvTqBtjmNk3No7v/dnP6Hr7wHxvCCRWubnmIfZ6QZAv2FU78pLX

8l3bsbmRAUg4UP9hH6ABVq4KQKMknxu1xQxLhpR1yIGPdiowMNTTrEG8cCx3w/w

==

</ds:X509Certificate>

</ds:X509Data>

</ds:KeyInfo>

</ds:Signature>

<Subject>

<NameID

Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">

scott@example.org

</NameID>

<SubjectConfirmation

Method="urn:oasis:names:tc:SAML:2.0:cm:bearer"/>

</Subject>

<Conditions NotBefore="2003-04-17T00:46:02Z"

NotOnOrAfter="2003-04-17T00:51:02Z">

<AudienceRestriction>

<Audience>http://www.opensaml.org/SP</Audience>

</AudienceRestriction>

</Conditions>

<AuthnStatement AuthnInstant="2003-04-17T00:46:00Z">

<AuthnContext>

<AuthnContextClassRef>

urn:oasis:names:tc:SAML:2.0:ac:classes>Password

</AuthnContextClassRef>

</AuthnContext>

</AuthnStatement>

</Assertion>

</Response>

6. SAML 과 XML 암호화 문법 및 처리

암호화는 기밀성을 구현하는 수단으로 사용된다. 기밀성을 필요로 하는 가장 일반적인 동기는 개개인들의 개인적인 프라이버시를 보호하거나 또는 경쟁에서의 이득 또는 유사한 이유들 때문에 조직의 비밀들을 보호하기 위한 것이다. 기밀성은 일부 다른 보호 메커니즘들의 효과를 보장하기 위해 또한 요구될 수 있다. 예를 들어, 비밀 암호나 키는 암호화될 수 있다.

SAML 주장의 전부 또는 일부를 기밀성을 가지도록 보호하기 위해 암호화를 사용하는 여러 방법들이 제공된다.

- 통신 기밀성은 특정 바인딩 또는 프로파일과 관련된 메커니즘들에 의해 보호될 수 있다. 예를 들어, SOAP 바인딩은 기밀성을 위하여 TLS 의 사용 또는 SOAP 메시지 보안 메커니즘들의 사용을 지원한다
- <SubjectConfirmation> 비밀은 <SubjectConfirmationData> 내에서 <ds:KeyInfo> 요소의 사용을 통해 보호될 수 있으며, 이것은 키들 또는 다른 비밀들이 암호화되는 것을 허용한다.
- 전체 <Assertion> 요소는 2.3.4 절에서 설명된 것처럼, 암호화될 수 있다.
- <BaseID> 또는 <NameID> 요소는 2.2.4 절에서 설명된 것처럼, 암호화될 수 있다.
- <Attribute> 요소들은 2.7.3.2 절에서 설명된 것처럼 암호화될 수 있다.

6.1. 일반적인 고려사항들(General Considerations)

<Assertion>, <BaseID>, <NameID> 그리고 <Attribute> 요소들의 암호화는 XML 암호화를 사용하여 제공된다. 암호화된 데이터와 선택적인 요소인 하나 또는 그 이상의 암호화된 키들은 평문 정보를 XML 인스턴스 내의 동일한 위치에서 대체해야만 한다. <EncryptedData> 요소의 Type 속성이 사용되어야 하며, 만약 이 속성이 존재하면, 반드시 <http://www.w3.org/2001/04/xmlenc#Element> 값을 가져야만 한다.

XML 암호화에서 사용될 목적으로 정의된 알고리즘들은 모두 암호화 작업을 수행하는데 사용될 수 있다. 암호화된 데이터를 포함해도 유효한 인스턴스가 되도록 SAML 스키마가 정의된다.

6.2. 서명들과 암호를 결합하기 (Combining Signatures and Encryption)

XML 암호화와 XML 서명의 사용은 결합될 수 있다. 주장이 서명되고 암호화될 때, 다음 규칙들이 적용된다. 의지하는 기관은 서명 및 암호화가 수행되었던 순서와는 반대로 서명 검증과 복호화를 수행되어야만 한다.

- 서명된 <Assertion> 요소가 암호화될 때, 서명이 먼저 계산되어 그 요소가 암호화되기 전에 <Assertion> 요소 내에 놓여져야만 한다.
- <BaseID>, <NameID> 또는 <Attribute> 속성이 암호화될 때, 암호화가 먼저 수행되고, 그리고 나서 암호화된 요소를 포함하는 주장 또는 메시지에 대하여 서명이 계산되어야만 한다.

7. SAML 확장성

SAML 은 주장과 프로토콜 스키마들을 확장하는 것을 포함하여, 여러 가지 방식으로 확장성을 지원한다. 새로운 프로파일을 어떻게 정의하는지에 대한 정보는 SAML 프로파일들 표준을 참조한다. 이 새로운 프로파일들은 SAML 프레임워크가 새로운 용도로 사용되기 위해 확장들과 결합될 수 있다.

7.1. 스키마 확장

SAML 스키마들에서는 요소들에 대한 치환(substitution)이 금지된다. 이것은 어떠한 SAML 요소들도 치환 그룹의 헤드요소(head element)로서 작용할 수는 없다는 것을 의미한다. 하지만, SAML 타입들은 final 로 정의되어 있지 않다. 그래서 모든 SAML 타입들은 확장되거나 제약될 수 있다. 실질적으로, 이것은 확장이 전형적으로 요소가 아닌 타입으로서만 정의가 되며 xsi:type 속성으로 SAML 인스턴스들에 포함된다는 것을 의미한다.

다음 절들은 확장성을 지원하도록 특수하게 설계되어온 요소들과 타입들만을 논의한다.

7.1.1. 주장 스키마 확장(Assertion Schema Extension)

SAML 주장 스키마는 만약 확장 메커니즘이 주장 패키지 또는 문장들에 대하여 사용된다면, 그것이 포함하는 주장 패키지와 문장들의 분리된 처리가 허용되도록 설계된다.

다음 요소들은 확장 스키마에서 확장 점들로 사용할 목적으로 특수하게 의도된 것이다. 그들의 타입들은 abstract 로 설정되며, 그래서 단지 유도된 타입들의 기반(base)으로만 사용될 수 있다.

- <BaseID>와 **BaseIDAbstractType**
- <Condition>과 **ConditionAbstractType**
- <Statement>와 **StatementAbstractType**

SAML 의 일부로써 직접적으로 사용될 수 있는 다음 구조들은 확장을 위해 특별히 관심 있는 목표들이다.

- <AuthnStatement>와 **AuthnStatementType**
- <AttributeStatement>와 **AttributeStatementType**
- <AuthzDecisionStatement>와 **AuthzDecisionStatementType**
- <AudienceRestriction>와 **AudienceRestrictionType**
- <ProxyRestriction>과 **ProxyRestrictionType**
- <OneTimeUse>와 **OneTimeUseType**

7.1.2. 프로토콜 스키마 확장(Protocol Schema Extension)

다음 SAML 프로토콜 요소들은 확장 스키마에서 확장 점들로 사용할 목적으로 특수하게 의도된 것이다. 그들의 타입들은 abstract 로 설정되며, 그래서 단지 유도된 타입들의 기반(base)으로만 사용될 수 있다.

- <Request>와 **RequestAbstractType**
- <SubjectQuery>와 **SubjectQueryAbstractType**

SAML 의 일부로써 직접적으로 사용될 수 있는 다음 구조들은 확장을 위한 특별히 관심 있는 목표들이다.

- <AuthnQuery>와 **AuthnQueryType**
- <AuthzDecisionQuery>와 **AuthzDecisionQueryType**
- <AttributeQuery>와 **AttributeQueryType**
- **StatusResponseType**

7.2. 스키마 와일드카드 확장 점들(Schema Wildcard Extension Points)

SAML 스키마들은 임의의 네임스페이스를 가지는 요소들과 속성들의 사용을 허용하기 위해 일부 지점들에 와일드카드 구조들을 사용하며, 이것은 확장 스키마를 필요로 하지 않으면서 내장된(built-in) 확장점으로 작용한다(servers).

7.2.1. 주장 확장 점들

주장 스키마에 있는 다음 구조들은 그들 내에 임의의 네임스페이스를 가지는 구조들이 존재하는 것을 허용한다.

- <SubjectConfirmationData>: xs:anyType 을 사용하고, 이것은 어떠한 하부-요소들과 속성들도 허용한다.
- <AuthnContextDecl>: xs:anyType 을 사용하며, 이것은 어떠한 하부-요소들과 속성들도 허용한다.
- <AttributeValue>: xs:anyType 을 사용하며, 이것은 어떠한 하부-요소들과 속성들도 허용한다.
- <Advice>과 AdviceType: SAML 에 고유한(SAML-native) 요소들뿐만 아니라, 느슨한 스키마 검증 처리(lax schema validation processing)를 가지고 다른 네임스페이스를 가지는 요소들을 허용한다.

주장 스키마 내의 다음 구조들은 임의의 전역 속성들을 허용한다.

- <Attribute>과 AttributeType

7.2.2. 프로토콜 확장 점들(Protocol Extension Points)

프로토콜 스키마에 있는 다음 구조들은 그들 내에 임의의 네임스페이스를 가지는 구조들이 존재하는 것을 허용한다.

- <Extensions>과 ExtensionsType: 느슨한 스키마 검증 처리(lax schema validation processing)를 가지고 다른 네임스페이스를 가지는 요소들을 허용한다.
- <StatusDetail>과 StatusDetailType: 느슨한 스키마 검증 처리(lax schema validation processing)를 가지고 다른 네임스페이스를 가지는 요소들을 허용한다.
- <ArtifactResponse>와 ArtifactResponseType: 느슨한 스키마 검증 처리(lax schema validation processing)를 가지고 어떠한 네임스페이스를 가지는 요소들도 허용한다(그렇지만, 이것은 특히 SAML 요청 또는 응답 메시지 요소를 운반하도록 의도된 것이다).

7.3. 식별자 확장(Identifier Extension)

SAML 은 상태 코드들, 이름 식별자 포맷 등과 같은 여러 가지 목적을 위하여 URI-기반 식별자들을 사용하며 이러한 목적으로 사용될 수 있는 일부 식별자들을 정의한다. 대부분의 식별자들은 8 장에 나열된다. 그러나, 항상 이러한 목적으로 추가적인 URI-기반 식별자들을 정의하는 것이 가능하다. 이들 추가적인 식별자들은 사용중인 프로파일 내에서 정의되도록 하는 것을 권고한다. 어떠한 경우에도 식별자로서 사용되는 주어진 URI 의 의미가 급격하게 변화해서는 안되며 또한 두 개의 다른 것들을 의미해서도 않된다.

8. SAML 에서 정의된 식별자들

다음 절은 공통 자원 접근 행동들, 주체 이름 식별자 포맷들 그리고 속성 이름 포맷들에 대한 URI-기반 식별자들을 정의한다.

가능하면, 이미 존재하는 URN 이 프로토콜을 명기하는데 사용된다. IETF 프로토콜들의 경우, 프로토콜을 명기하는 가장 최신의 RFC 에 대한 URN 이 사용된다. SAML 을 위해 특별하게 생성된 URI 참조들은, 그들이 처음 도입된 곳의 표준 집합 버전에 따라, 다음 계통들(stems) 중에 하나를 가진다.

```
urn:oasis:names:tc:SAML:1.0:
urn:oasis:names:tc:SAML:1.1:
urn:oasis:names:tc:SAML:2.0:
```

8.1. 행동 네임스페이스 식별자들(Action Namespace Identifiers)

다음 식별자들이 자원들에 대하여 수행되는 공통적인 행동들의 집합을 참조하는 <Action> 요소의 Namespace 속성에서 사용될 수 있다.

8.1.1. Read/Write/Execute/Delete/Control

URI: urn:oasis:names:tc:SAML:1.0:action:rwedc

정의된 행동들:

Read Write Execute Delete Control

이들 행동들은 다음과 같이 해석된다.

Read

주체가 자원을 읽을 수 있다.

Write

주체가 자원을 변경할 수 있다.

Execute

주체가 자원을 실행할 수 있다.

Delete

주체가 자원을 삭제할 수 있다.

Control

주체가 자원에 대한 접근 제어 정책을 명기할 수 있다.

8.1.2. Read/Write/Execute/Delete/Control with Negation

URI: urn:oasis:names:tc:SAML:1.0:action:rwedc-negation

정의된 행동들:

Read Write Execute Delete Control ~Read ~Write ~Execute ~Delete ~Control

8.1.1 절에서 명기된 행동들은 거기서 설명된 것과 동일한 방식으로 해석된다. “~” 가 접두사로 붙은 행동들은 부정된 허가들(negated permissions)이고 기술된(stated) 허가가 부정되었다는 것을 단정적으로 명기하는데 사용된다. 따라서, 행동 ~Read 를 수행하도록 인가된 것으로 설명된 주체는 단정적으로 읽기 허가가 부인된다.

SAML 기관은 하나의 행동과 그것의 부정된(negated) 형태 두 가지 모두를 인가하지 않아야만 한다.

8.1.3. Get/Head/Put/Post

URI: urn:oasis:names:tc:SAML:1.0:action:ghpp

정의된 행동들:

GET HEAD PUT POST

이들 행동들은 대응되는 HTTP 연산들에 바인드된다. 예를 들어, 자원에 대해 GET 행동을 수행하는 것이 인가된 주체는 자원을 검색하는 것이 인가된다.

GET 과 HEAD 행동들은 전통적인 읽기 허가와 느슨하게 대응되며, PUT 과 POST 행동들은 쓰기 허가와 느슨하게 대응된다. 그러나 HTTP GET 연산은 데이터가 수정되도록 할 수도 있고 POST 연산은 요청에서 명기된 것이 아닌 다른 자원에 대해 변경을 일으킬 수 있기 때문에, 이러한 대응이 정확한 것은 아니다. 이러한 이유 때문에, 별도의 Action URI 참조들이 제공된다.

8.1.4. UNIX File Permissions

URI: urn:oasis:names:tc:SAML:1.0:action:unix

정의된 행동들은 수치화된(8 진) 표기법으로 표시된 유닉스 파일 접근 허가들의 집합이다.

행동 문자열은 4-숫자 (four digit) 수치 코드이다.

extended user group world

확장된 접근 허가가 값을 가지는 곳에서는

만약 sgid 가 설정되면, +2

만약 suid 가 설정되면, +4

사용자, 그룹과 세계 접근 허가들이 값을 가지는 곳에서

만약 실행 허가가 주어진다면, +1

만약 쓰기 허가가 주어진다면, +2

만약 읽기 허가가 주어진다면, +4

예를 들어, 0754 는 다음과 같은 유닉스 파일 접근 허가를 표시한다: 사용자 읽기, 쓰기 그리고 실행하기; 그룹 읽기 그리고 실행하기; 그리고 세계 읽기

8.2. 속성 이름 포맷 식별자들(Attribute Name Format Identifiers)

다음 식별자들이 이름을 해석할 목적으로 속성 이름의 분류(classification)를 참조하기 위해 **AttributeType** 복합 타입에 정의된 NameFormat 속성에서 사용될 수 있다.

8.2.1. Unspecified

URI: urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified

이 속성의 해석은 개별적인 구현들에 달려있다.

8.2.2. URI Reference

URI: urn:oasis:names:tc:SAML:2.0:attrname-format:uri

이 속성은, 예를 들어 XACML 속성 식별자들에서 사용되는 것과 같이, URI 참조들에 대한 관례(convention)을 따른다. URI 내용의 해석 또는 네이밍 스키마(naming schema)는 응용에 따라 고유하다(application-specific).

8.2.3. Basic

URI: urn:oasis:names:tc:SAML:2.0:attrname-format:basic

속성 이름으로써 받아들일 수 있는 문자열들의 클래스는 W3C XML Datatypes 3.3.6 절에 정의된 프리미티브 타입(primitive type) **xs:Name** 에 속하는 값들의 집합으로부터 유도되어야만 한다.

8.3. 이름 식별자 포맷 식별자들(Name Identifier Format Identifiers)

다음 식별자들은 요소들의 내용에 대한 공통된 포맷들과, 만약 있다면, 연관된 처리 규칙들을 참조하기 위해 <NameID>, <NameIDPolicy> 또는 <Issuer> 요소의 Format 속성에서 사용될 수 있다.

주의: SAML 1.1 에서 비난 받은 여러 식별자들이 SAML 2.0 에서 제거되었다.

8.3.1. Unspecified

URI: urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified

이 요소의 해석은 개별적인 구현들에 달려있다.

8.3.2. Email Address

URI: urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress

이 요소의 내용이 email 주소 형식이며, 특히 IETF RFC 2822 3.4.1 절에서 정의된 것 같은 “addr-spec” 형식임을 가리킨다. addr-spec 은 local-part@domain 형식을 가진다. addr-spec 은 그것 앞에 (공통 이름과 같은) 구(phrase)를 가지지 않으며, 그것 뒤에 (괄호로 둘러싸인 텍스트인) 주석을 가지고 있지 않으며, “<”과 “>”으로 둘러싸이지 않는다.

8.3.3. X.509 Subject Name

URI: urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName

이 요소의 내용이 W3C XML Signature 의 <ds:X509SubjectName> 요소의 내용을 위해 명기된 형식임을 가리킨다. 구현자들은 XML 서명 표준이 IETF RFC 2253 에서 주어진 규칙들과 다른 X.509 주체에 대한 인코딩 규칙들을 명기한다는 것에 주의해야 한다.

8.3.4. Windows Domain Qualified Name

URI: urn:oasis:names:tc:SAML:1.1:nameid-format:WindowsDomainQualifiedName

이 요소의 내용이 윈도우 도메인 한정 이름(Windows Domain Qualified Name)이라는 것을 가리킨다. 윈도우 도메인 한정 이름은 “도메인이름\사용자이름” 형태의 문자열이다. 도메인 이름과 “\” 분리자는 생략될 수 있다.

8.3.5. Kerberos Principal Name

URI: urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos

이 요소의 내용이 name[/instance]@REALM 형식을 사용하는 Kerberos principal 이름 형식이라는 것을 가리킨다. 문법, 이름에 대하여 허용된 포맷과 문자들, 인스턴스 그리고 영역(realm)은 IETF RFC 1510 에서 설명된다.

8.3.6. Entity Identifier

URI: urn:oasis:names:tc:SAML:2.0:nameid-format:entity

이 요소의 내용이 (SAML 기관, 요청자 또는 응답자 등과 같은) SAML-기반 서비스들을 제공하는 엔티티의 식별자이거나 또는 (브라우저 SSO 프로파일을 지원하는 서비스 제공자와 같은) SAML 프로파일들에서 참여자인 엔티티의 식별자라는 것을 가리킨다. 이와 같은 식별자는 SAML 요청, 응답 또는 주장의 발급자를 식별하는 <Issuer> 요소에서 사용되거나 SAML 요청들, 응답들 그리고 주장들을 발급할 수 있는 시스템 엔티티들에 대해 주장하는 <NameID> 요소 내에서 사용될 수 있다. 이것은 또한 그것의 목적이 다양한 프로토콜 교환에서 시스템 엔티티를 식별하는 것인 다른 요소들과 속성들에서 사용될 수 있다.

이와 같은 식별자의 문법은 길이가 1024 문자를 넘지 않는 URI 이다. 시스템 엔티티는 자신을 식별하기 위해 자신이 소유한 도메인 이름을 포함하는 URI 를 사용하는 것이 권고된다.

NameQualifier, SPNameQualifier, 그리고 SPProvidedID 속성들은 생략되어야만 한다.

8.3.7. Persistent Identifier

URI: urn:oasis:names:tc:SAML:2.0:nameid-format:persistent

이 요소의 내용이 IdP 와 서비스 제공자 또는 서비스 제공자들 제휴그룹에 특수한 (specific) 사용자에 대한 지속적(persistent)이고 불투명한 식별자(persistent opaque identifier)임을 가리킨다. IdP 들에 의해 생성된 지속적인 이름 식별자들은 (예를 들어, 사용자 이름과 같은) 주체의 실질적인 식별자와 어떠한 인식할수 있는 대응 관계를 가지지 않는(no discernible correspondence) 의사-랜덤(pseudo-random) 값들을 사용하여 구성되어야만 한다. 이것의 의도는 주체의 신원 또는 활동들에 대한 발견(discovery)을 방지하는 비-공개, 쌍방향 의사명(pair-wise pseudonym)을 생성하는 것이다. 지속적인 이름 식별자 값들은 256 문자 길이를 초과하지 않아야만 한다.

이 요소의 NameQualifier 는, 만약 존재한다면, 반드시 이름 식별자를 생성하는 IdP 의 유일한 식별자를 포함해야만 한다. (8.3.6 절을 참조). 만약, 프로토콜 메시지의 발급자 또는 그것의 주체에 식별자를 포함하는 주장과 같이, 그 값이 요소를 포함하는 메시지 문맥으로부터 유도될 수 있다면, NameQualifier 는 생략될 수 있다. 다른 시스템 엔티티가 나중에 식별자를 포함하는 자신의 프로토콜 메시지 또는 주장을 발급할 수 있다는 것에 주의해야 한다; NameQualifier 속성은 이 경우에 변경되지 않지만 반드시 원래 식별자를 생성한 엔티티를 계속해서 식별해야만 한다. 그리고 이와 같은 경우에는 생략되지 않아야만 한다.

요소의 SPNameQualifier 속성은, 만약 존재하면, 이 식별자가 생성의 대상이 되는 서비스 제공자 또는 제공자들의 제휴기관의 유일한 식별자를 포함해야만 한다 (8.3.6 절을 참조). 만약 요소가 서비스 제공자에 의해 직접적으로 소비될 것만을 목적으로 하는 메시지에 포함되면, 이것은 생략될 수 있고, 그리고 그 값은 서비스 제공자의 유일한 식별자일 것이다.

요소의 SPProvidedID 속성은, 만약 있다면 (3.6 절을 참조), 서비스 제공자 또는 제휴기관에 의해 가장 최신에 설정된 사용자의 다른 식별자를 포함해야만 한다. 만약 이러한 식별자가 설정되어 있지 않으면, 그러면 이 속성은 반드시 생략되어야만 한다.

지속적인 식별자들은 프라이버시 보호 메커니즘으로써 의도된다. 그렇게 이들 식별자들은 공유된 식별자를 설정한 제공자가 아닌 다른 제공자들에게 평문(clear text)로 공유되지 않아야만 한다. 더욱이, 그들은 적절한 제어와 보호 조치 없이 로그 파일들이나 유사한 위치들에 나타나지 않아야만 한다. 이와 같은 요구사항이 없는 배치들은 그들의 SAML 교환에 있어 다른 종류의 식별자들을 자유롭게 사용하지만 지속적이지만 투명한(non-opaque) 값들을 가지고 이러한 포맷을 오버로드(overload) 하지 않아야만 한다.

또한, 지속적인 식별자들은 전형적으로 한쌍의 제공자들 사이에서의 계정 연결 관계(account linking relationship)를 반영하는데 사용되는 반면, 서비스 제공자는 지속적인 식별자의 긴 기간 특성(long term nature)를 인식하거나 또는 이용하거나, 또는 이와 같은 연결을 설정할 의무를 가지지는 않는다. 이와 같은 일방향(“one-sided”) 관계는 인식할수 있을 정도로 다르지 않으며 IdP 의 행동 또는 이 표준에서 정의된 프로토콜들 내의 지속적인 식별자들에 특수한 어떠한 처리 규칙들에도 영향을 주지 않는다.

마지막으로, NameQualifier 와 SPNameQualifier 속성들은 사용되지는 않지만 생성의 방향을 가리킨다는 사실에 주의해야 한다. 만약 지속적인 식별자가 특정한 IdP 에 의해 생성되면, NameQualifier 속성 값은 그 시점에 영구히 설정된다. 만약 이와 같은 식별자를 수신한 서비스 제공자가 IdP 의 역할을 취하고 그 식별자를 포함하는 자신의 주장을 발급하면, NameQualifier 속성 값은 변하지 않는다 (그리고 물론 생략되지도 않을 것이다). 또 다른 방식으로, 사용자를 나타내는 자신의 지속적인 식별자를 선택하여 두 값을 연결하는 것을 선택할 수 도 있다. 이것은 배치에서 결정할 문제이다.

8.3.8. Transient Identifier

URI: urn:oasis:names:tc:SAML:2.0:nameid-format:transient

이 요소의 내용이 일시적인(transient) 의미를 가지는 식별자이며 의지하는 기관에서 불투명하고 임시 값으로 취급해야 한다는 사실을 가리킨다. 일시적인 식별자 값들은 SAML 식별자들에 대한 규칙에 따라 생성되어야만 하고 (1.3.4 절을 참조) 길이가 256 문자를 넘어서지 않아야만 한다.

NameQualifier 와 SPNameQualifier 속성들은 식별자가 일시적이고 임시적인 쌍방향 식별자를 나타낸다는 것을 의미하는데 사용될 수 있다. 이와 같은 경우, 그들은 8.3.7 절에서 명기된 규칙들에 따라 생략될 수 있다.

8.4. 동의 식별자들(Consent Identifiers)

다음 식별자들은 메시지에 대하여 어떤 조건하에서 사용자가 동의를 제공했는지 또는 하지 않았는지에 대한 내용을 통신하기 위해 **RequestAbstractType** 과 **StatusResponseType** 복합 타입들에 정의된 Consent 속성에서 사용될 수 있다.

8.4.1. Unspecified

URI: urn:oasis:names:tc:SAML:2.0:consent:unspecified

사용자 동의에 대한 어떠한 내용도 명기되어 있지 않다.

8.4.2. Obtained

URI: urn:oasis:names:tc:SAML:2.0:consent:obtained

사용자의 동의가 메시지 발급자에 의해 얻어졌다는 것을 가리킨다.

8.4.3. Prior

URI: urn:oasis:names:tc:SAML:2.0:consent:prior

사용자의 동의가 메시지를 시작하는(initiate)하는 행동이 일어나기 전에 메시지 발급자에 의해 얻어졌다는 것을 가리킨다.

8.4.4. Implicit

URI: urn:oasis:names:tc:SAML:2.0:consent:current-implicit

사용자의 동의가, 메시지를 시작하는 행동을 하는 동안에, 더 넓은 동의의 표시의 일부로써, 메시지 발급자에 의해 묵시적으로(implicitly) 얻어졌다는 것을 가리킨다. 묵시적인 동의는 전형적으로 활동들의 세션 일부와 같이 prior consent 보다 시간적으로 행동에 더 가까우며 동의의 제출도 더 가깝다.

8.4.5. Explicit

URI: urn:oasis:names:tc:SAML:2.0:consent:current-explicit

사용자의 동의가 메시지를 구성하는 행동을 하는 동안에, 명시적으로 얻어졌다는 것을 가리킨다.

8.4.6. Unavailable

URI: urn:oasis:names:tc:SAML:2.0:consent:unavailable

메시지의 발급자가 동의를 얻을 수 없다는 것을 가리킨다.

8.4.7. Inapplicable

URI: urn:oasis:names:tc:SAML:2.0:consent:inapplicable

메시지 발급자는 자신이 사용자 동의를 얻거나 또는 보고할 필요가 없다고 믿는다는 것을 가리킨다.

표준 작성 공헌자

표준 번호 : TTAS.IT-X1141_1

이 표준의 제정·개정 및 발간을 위해 아래와 같이 여러분들이 공헌하셨습니다.

구분	성명	위원회 및 직위	연락처	소속사
과제 제안	조영섭	PG101 위원	042-860-6942 yscho@etri.re.kr	ETRI
표준 초안 제출	조영섭	PG101 위원	042-860-6942 yscho@etri.re.kr	ETRI
표준 초안 검토 및 작성	이석래	PG101 의장	02-405-5330 sllee@kisa.or.kr	KISA
	진승헌	PG101 부의장	042-860-1254 jinsh@etri.re.kr	ETRI
	백종현	PG101 간사	02-405-5423 jhbaek@kisa.or.kr	KISA
	조상래	연구원	042-860-6939 slcho@etri.re.kr	ETRI
		외 PG101 위원		
표준안 심의	정교일	공통기반기술위원회 의장	042-860-1920 kyoil@etri.re.kr	ETRI
	원유재	공통기반기술위원회 부의장	02-405-5360 yjwon@kisa.or.kr	KISA
	이필중	공통기반기술위원회 부의장	054-279-2232 pjl@postech.ac.kr	포항공대
	김응배	공통기반기술위원회 부의장	042-860-5296 ebkim@etri.re.kr	ETRI
		외 TC1 위원		
사무국 담당	김 선	팀 장	031-724-0080 skim@tta.or.kr	TTA
	오흥룡	과 장	031-724-0083 hroh@tta.or.kr	TTA



정보통신단체표준(국문표준)

SAML 2.0 주장과 프로토콜
(SAML 2.0 Assertions and Protocols)

발행인 : 한국정보통신기술협회 회장

발행처 : 한국정보통신기술협회

463-824, 경기도 성남시 분당구 분당로 47

Tel : 031-724-0114, Fax : 031-724-0109

발행일 : 2006.12.
