1

2 Web Services Security:

3 UsernameToken Profile 1.0

4 Errata 1.0

5 Committee Draft 200401, September 2004

6 **Document identifier:**
7 {*WSS: SOAP Message Security }-{UsernameToken Profile*}-{*1.0*} (Word) (PDF)

8 **Document Location:**
9 http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0-
10 errata-003

11 **Errata Location:**
12 http://www.oasis-open.org/committees/wss

13 **Editors:**

Anthony	Nadalin	IBM
Phil	Griffin	Individual
Chris	Kaler	Microsoft
Phillip	Hallam-Baker	VeriSign
Ronald	Monzillo	Sun

14 **Contributors:**

Gene	Thurston	AmberPoint
Frank	Siebenlist	Argonne National Lab
Merlin	Hughes	Baltimore Technologies
Irving	Reid	Baltimore Technologies
Peter	Dapkus	BEA
Hal	Lockhart	BEA
Symon	Chang	CommerceOne
Srinivas	Davanum	Computer Associates
Thomas	DeMartini	ContentGuard
Guillermo	Lao	ContentGuard
TJ	Pannu	ContentGuard
Shawn	Sharp	Cyclone Commerce
Ganesh	Vaideeswaran	Documentum
Sam	Wei	Documentum
John	Hughes	Entegrity
Tim	Moses	Entrust
Toshihiro	Nishimura	Fujitsu

Tom	Rutt	Fujitsu
Yutaka	Kudo	Hitachi
Jason	Rouault	HP
Paula	Austel	IBM
Bob	Blakley	IBM
Joel	Farrell	IBM
Satoshi	Hada	IBM
Maryann	Hondo	IBM
Michael	McIntosh	IBM
Hiroshi	Maruyama	IBM
David	Melgar	IBM
Anthony	Nadalin	IBM
Nataraj	Nagaratnam	IBM
Wayne	Vicknair	IBM
Kelvin	Lawrence	IBM (co-Chair)
Don	Flinn	Individual
Bob	Morgan	Individual
Bob	Atkinson	Microsoft
Keith	Ballinger	Microsoft
Allen	Brown	Microsoft
Paul	Cotton	Microsoft
Giovanni	Della-Libera	Microsoft
Vijay	Gajjala	Microsoft
Johannes	Klein	Microsoft
Scott	Konersmann	Microsoft
Chris	Kurt	Microsoft
Brian	LaMacchia	Microsoft
Paul	Leach	Microsoft
John	Manferdelli	Microsoft
John	Shewchuk	Microsoft
Dan	Simon	Microsoft
Hervey	Wilson	Microsoft
Chris	Kaler	Microsoft (co-Chair)
Prateek	Mishra	Netegrity
Frederick	Hirsch	Nokia
Senthil	Sengodan	Nokia
Lloyd	Burch	Novell
Ed	Reed	Novell
Charles	Knouse	Oblix
Steve	Anderson	OpenNetwork (Sec)
Vipin	Samar	Oracle
Jerry	Schwarz	Oracle
Eric	Gravengaard	Reactivity
Stuart	King	Reed Elsevier
Andrew	Nash	RSA Security
Rob	Philpott	RSA Security
Peter	Rostin	RSA Security
Martijn	de Boer	SAP
Blake	Dournaee	Sarvega
Pete	Wenzel	SeeBeyond
Jonathan	Tourzan	Sony
Yassir	Elley	Sun Microsystems

Jeff	Hodges	Sun Microsystems
Ronald	Monzillo	Sun Microsystems
Jan	Alexander	Systinet
Michael	Nguyen	The IDA of Singapore
Don	Adams	TIBCO
John	Weiland	US Navy
Phillip	Hallam-Baker	VeriSign
Mark	Hays	Verisign
Hemma	Prafullchandra	VeriSign

Abstract:

This document contains a list of errata against WSS Username Token Profile 1.0 that have been approved by the WSS Technical Committee.

Status:

This version of the errata is a working draft of the committee. As such, it may change prior to incorporation into a future OASIS Standard. Please send comments to the editors. If you are on the wss@lists.oasis-open.org list for committee members, send comments there. If you are not on that list, subscribe to the wss-comment@lists.oasis-open.org list and send comments there. To subscribe, send an email message to wss-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message. For patent disclosure information that may be essential to the implementation of this specification, and any offers of licensing terms, refer to the Intellectual Property Rights section of the OASIS Web Services Security Technical Committee (WSS TC) web page at http://www.oasis-open.org/committees/wss/ipr.php. General OASIS IPR information can be found at http://www.oasis-open.org/who/intellectualproperty.shtml.

Table of Contents

1 Issues Addressed

The following issues have been addressed in this document:

ISSUE	DESCRIPTION
259	Editorial comments on Username Token profile - post v1 review period.

2 Typographical Errors

None

3 Normative Errors

3.1 Section 2.2 Namespaces

Add the following after line 81
URI fragments defined in WSS: Username Token Profile 1.0 are relative to a base URI of `http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0`

3.2 Section 3.1 Usernames and Passwords

Delete the following line (89):
Passwords of type wsse:PasswordText and wsse:PasswordDigest are not limited to
and replace it with:
Passwords of type PasswordText and PasswordDigest are not limited to

Delete the following lines (91-93):
Having a type of wsse:PasswordText, wsse:PasswordDigest merely implies that the information held in the password is "in the clear"…
and replace it with:
Having a type of PasswordText merely implies that the information held in the password is "in the clear" …

Delete the following line (98):
Passwords of type wsse:PasswordText and wsse:PasswordDigest are defined as being
and replace it with:
Passwords of type PasswordDigest are defined as being

Delete the following line (101-102):
the digest offers no real additional security over use of wsse:PasswordText and wsse:PasswordDigest.
and replace it with:
the digest offers no real additional security over use of PasswordText.

Delete the following line (132):
Note that `wsse:PasswordDigest` can only be used if the plain text password (or password
and replace it with:
Note that `PasswordDigest` can only be used if the plain text password (or password

Delete the following line(235):
"ValueType attribute is not required. If specified, the value of <wsse:UsernameToken> MUST"
and replace it with:
"ValueType attribute is not required. If specified, the value of #UsernameToken MUST"

3.3 Section 3.2 Token Reference

Delete the following line(235):
`ValueType` attribute is not required. If specified, the value of `<wsse:UsernameToken>` MUST
and replace it with:

`ValueType` attribute is not required. If specified, the value of `#UsernameToken` MUST

3.4 Section 3.4 Key Derivation (new)

Add the following as new section after line 257:

The password associated with a username may be used to derive a shared secret key for the purposes of integrity or confidentiality protecting message contents. This section defines schema extensions and a procedure for deriving such keys. This procedure MUST be employed when keys are to be derived from passwords in order in insure interoperability.

It must be noted that passwords are subject to several kinds of attack, which in turn will lead to the exposure of any derived keys. This key derivation procedure is intended to minimize the risk of attacks on the keys, to the extent possible, but it is ultimately limited by the insecurity of a password that it is possible for a human being to remember and type on a standard keyboard. This is discussed in more detail in the security considerations section of this document.

Two additional elements are required to enable to derivation of a key from a password. They are `<wsse:Salt>` and `<wsse:Iteration>`. These values are not secret and MUST be conveyed in the Username token when key derivation is used. When key derivation is used the password MUST NOT be included in the Username token. The receiver will use its knowledge of the password to derive the same key as the sender.

The following illustrates the syntax of the `<wsse:Salt>` and `<wsse:Iteration>` elements.

```
<wsse:UsernameToken wsse:Id="…">
    <wsse:Username>…</wsse:Username>
    <wsse:Salt>…</wsse:Salt>
    <wsse:Iteration>…</wsse:Iteration>
</wsse:UsernameToken>
```

The following describes these elements.

/wsse:UsernameToken/wsse:Salt

This element is combined with the password as described below. Its value is a 128 bit number expressed in hexadecimal. It MUST be present when key derivation is used.

/wsse:UsernameToken/wsse:Iteration

This element indicates the number of times the hashing operation is repeated when deriving the key. It is expressed as a decimal value. If it is not present, a value is 1000 is used for the iteration count.

A key derived from a password may be used either in the calculation of a Message Authentication Code (MAC) or as a symmetric key for encryption. When used in a MAC, the key length will always be 160 bits. When used for encryption, an encryption algorithm MUST NOT be used which requires a key of length greater than 160 bits. A sufficient number of the high order bits of the key will be used for encryption. Unneeded low order bits will be discarded. For example, if the AES-128 algorithm is used, the high order 128 bits will be used and the low order 32 bits will be discarded from the derived 160 bit value.

The `<wsse:Salt>` element is constructed as follows. The high order 8 bits of the Salt will have the value of 01 if the key is to be used in a MAC and 02 if the key is to be used for encryption. The remaining 120 low order bits of the Salt should be a random value.

The key is derived as follows. The password and Salt are concatenated in that order. Only the actual octets of the password are used, it is not padded or zero terminated. This value is hashed using the SHA1 algorithm. The result of this operation is also hashed using SHA1. This process is repeated until the total number of hash operations equals the Iteration count.

In other words: K1 = SHA1(password + Salt)

K2 = SHA1(K1)

…

Kn = SHA1 (Kn-1)

Where + means concatenation and n is the iteration count.

The resulting 160 bit value is used in a MAC function or truncated to the appropriate length for encryption.

3.5 Section 4 Security Concideration

Add after line 282:

The security of keys derived from passwords is limited by the attacks available against passwords themselves, such as guessing and brute force. Because of the limited size of password that human beings can remember and limited number of octet values represented by keys that can easily be typed, a typical password represents the equivalent of an entropy source of a maximum of only about 50 bits. For this reason a maximum key size of only 160 bits is supported. Longer keys would simply increase processing without adding to security.

The key derivation algorithm specified here is based on one described in RFC 2898. It is referred to in that document as PBKDF1. It is used instead of PBKDF2, because it is simpler and keys longer than 160 bits are not required as discussed previously.

The purpose of the salt is to prevent the bulk pre-computation of key values to be tested against distinct passwords. The Salt value is defined so that MAC and encryption keys are guaranteed to have distinct values even when derived from the same password. This prevents certain cryptanalytic attacks.

The iteration count is intended to increase the work factor of a guessing or brute force attack, at a minor cost to normal key derivation. An iteration count of at least 1000 (the default) SHOULD always be used.

4 Non-Normative Errors

None

177 # 5 Clarifications

178
179 The following table lists the full URI for each URI fragment referred to in the specification.

URI Fragment	Full URI
#PasswordDigest	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordDigest
#PasswordText	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText
#UsernameToken	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0 #UsernameToken

180

Appendix A: Revision History

Rev	Date	What
1	06/25/04	First Draft of Errata
2	07/06/04	Updated per comments on list

This section is non-normative.

Appendix B: Notices

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification, can be obtained from the OASIS Executive Director.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS Executive Director.

This section is non-normative.