~~Two~~ Three optional elements are introduced in the <wsse:UsernameToken> element to provide a countermeasure for replay attacks: <wsse:Nonce>~~ and,~~ <wsu:Created> and <wsse:Destination>. A nonce is a random value that the sender creates to include in each Username token that it sends. Although using a nonce is an effective countermeasure against replay attacks, it requires a server to maintain a cache of used nonces, consuming server resources. Combining a nonce with a creation timestamp has the advantage of allowing a server to limit the cache of nonces to a "freshness" time period, establishing a bound on resource requirements. Combining a nonce with a destinationDomain ensures that the <wsse:UsernameToken> element cannot be replayed to a different server. If ~~either or both~~any of <wsse:Nonce>~~ and,~~ <wsu:Created> and <wsse:Destination are present they must be included in the digest value as follows:

Password_Digest = Base64 (SHA -1 (nonce + created + destination + password))

That is, concatenate the nonce, creation timestamp, destination and the password (or shared secret or password equivalent), digest the combination using the SHA -1 hash algorithm, then include the Base64 encoding of that result as the Password (digest). This helps obscure the password and offers a basis for preventing replay attacks. For web service providers to effectively thwart replay attacks, three counter measures are recommended:

1. First, it is recommended that web service providers reject any UsernameToken *not* using *both* nonce *and* creation timestamps
2. Second, it is recommended that web service producers provide a timestamp “freshness” limitation, and that any UsernameToken with “stale” timestamps berejected. As a guideline, a value of five minutes can be used as a minimum to detect, and thus reject, replays.
3. Third, it is recommended that used nonces be cached for a period at least as long s the timestamp freshness limitation period, above, and that UsernameTokens with nonces that have already been used (and are thus in the cache) be rejected

Note that the nonce is hashed using the octet sequence of its decoded value while the timestamp and destinartion are ~~is~~ hashed using the octet sequences of ~~its~~ their UTF8 encoding as specified in the contents of the element.

Note that passwords of either type (wsse:PasswordText or wsse:PasswordDigest) can only be used if the plain text password (or password equivalent) is available to both the requestor and the recipient.
The following illustrates the XML [2] syntax of this element:

```
<wsse:UsernameToken wsu:Id="Example-1">
   <wsse:Username> ... </wsse:Username>
   <wsse:Password Type="..."> ... </wsse:Password>
   <wsse:Nonce EncodingType="..."> ... </wsse:Nonce>
   <wsse:Destination> ... </wsse:Destination>
   <wsu:Created> ... </wsu:Created>
</wsse:UsernameToken>
```

The following describes the attributes and elements listed in the example above:

/wsse:UsernameToken/Password

This optional element provides password information (or equivalent such as a hash). It isrecommended that this element only be passed when a secure transport is being used.

/wsse:UsernameToken/Password/@Type

This optional attribute specifies the type of password being provided. The following table identifies the pre-defined types:

Value	**Description**
wsse:PasswordText (default)	The actual password for the username, the password hash, or derived password or S/KEY.

wsse:PasswordDigest The digest of the password (and optionally nonce and/or creation timestame) for the username using the algorithm described above.

/wsse:UsernameToken/Password/@{any}

This is an extensibility mechanism to allow additional attributes, based on schemas, to be added to the element.

/wsse:UsernameToken/wsse:Nonce

This optional element specifies a cryptographically random nonce. Each message including a Nonce element should use a new nonce value in order for web service providers to detect replay attacks

/wsse:UsernameToken/wsse:Nonce/@EncodingType

This optional attribute specifies the encoding type of the nonce (see the definition of <wsse:BinarySecurityToken> for valid values). If this attribute isn't specified then the default of Base64 encoding is used.

/wsse:UsernameToken/wsu:Created

This optional <wsu:Created> element specifies a timestamp used to indicate the creation time. It is defined as part of the <wsu:Timestamp> definition.

/wsse:UsernameToken/wsse:Destination

This optional element is a URI specifying the destination domain. The provider should own this domain and should be capable of ensuring that nonces are shared between all systems that share the domain. This value would typically be the URL to which a message was sent

All compliant implementations must be able to process the `<wsse:UsernameToken>` element. The following example illustrates the use of this element. In this example the password is sent as clear text and therefore this message should be sent over a confidential channel:

```
<S:Envelope xmlns:S="http://www.w3.org/2001/12/soap-envelope"
   xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext">
   <S:Header>
      ...
      <wsse:Security>
         <wsse:UsernameToken>
            <wsse:Username>Zoe</wsse:Username>
            <wsse:Password>IloveDogs</wsse:Password>
         </wsse:UsernameToken>
      </wsse:Security>
      ...
   </S:Header>
   ...
</S:Envelope>
```

The following example illustrates using a digest of the password along with a nonce and creation timestamp:

```
<S:Envelope xmlns:S="http://www.w3.org/2001/12/soap-envelope"
   xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext">
   <S:Header>
      ...
      <wsse:Security>
         <wsse:UsernameToken
            xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext"
```

```
            xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/utility">
            <wsse:Username>NNK</wsse:Username>
            <wsse:Password Type="wsse:PasswordDigest">
               weYI3nXd8LjMNVksCKFV8t3rgHh3Rw==
            </wsse:Password>
            <wsse:Nonce>WScqanjCEAC4mQoBE07sAQ==</wsse:Nonce>
            <wsu:Created>2003-07-16T01:24:32Z</wsu:Created>
            <wsse:Destination>http://my.company.com</wsse:Destination>
        </wsse:UsernameToken>
      </wsse:Security>
      ...
   </S:Header>
   ...
</S:Envelope>
```