版本信息

日期	版本	描述	作者
2004-04-13	v1.0	最初版本	LDAPChina.com

Authentication Methods for LDAP

LDAP 认证方法

本备忘录的状态（Status of this Memo）

本文档定义了一个用于 Internet 通讯的 Internet 标准跟踪协议，为了发展的需要讨论和建议。对于这个协议的状况和地位请参照 Internet 官方协议标准（"Internet Official Protocol Standards"（STD 1））的当前版。这个备忘录的传播是不受限制的。

版权提示（Copyright Notice）

目 录

摘要（Abstract）

本文档描述了需要并且推荐在 LDAP 服务器实现中使用的安全机制的特定组合。

1、介绍（Introduction）

LDAP 第 3 版是一个强大的目录访问协议。

它提供了查询、接收和操作目录内容的方法，以及访问大量安全功能的方法。

为了在 Internet 上得到最好的运用，这些安全功能具备互操作性是必需的，所以必须有一个安全功能的最小子集，该安全功能子集对所有声称符合 LDAPv3 的服务器实现都通用。

对 LDAP 目录服务最基本的威胁包括：

1、 利用数据接收（data-fetch）操作对数据进行未授权访问；

2、 利用监听其它访问，来重用客户端认证信息，进行未授权访问；

3、 利用监听其它访问，对数据进行未授权访问；

4、 未授权的修改数据；

5、 未授权的修改配置；

6、 未授权地使用资源或过度使用资源（拒绝服务）；

7、 欺骗目录：使用欺骗手段让客户端相信信息来自于目录，其实目录并没有返回任何信息，或者修改在传输中的数据，或者让客户端失去连接。

第 1、4、5 和第 6 个威胁来自于有敌意的客户端。第 2、3 和第 7 个威胁来自于客户端和服务器路径之间的有敌意的代理，或者来自于一个伪装的服务器。

LDAP 协议体系可以使用下列安全机制加以保护：

1、 使用 SASL（参考文档[2]）机制集进行客户端认证，这种机制可能会使用 TLS 凭证交换机制（TLS credentials exchange mechanism）；

2、 使用基于请求者的已认证身份的访问控制方法进行客户端认证；

3、 使用 TLS 协议或者数据完整性（data-integrity）SASL 机制的方法保护数据完整性；

4、 使用 TLS 协议或者数据加密（data-encrypting）SASL 机制的方法避免窃听；

5、 使用对服务 control（控制）进行管理限制的方法对资源的使用进行限制；

6、 使用 TLS 协议或 SASL 机制进行服务器认证。

此刻，使用 LDAP 协议范围之外的方法强迫应用访问控制。

在本文档中，术语"user"表示所有使用目录检索或存储信息的 LDAP 客户端应用程序。

在本文档中使用的关键字"MUST"，"MUST NOT"，"REQUIRED"，"SHALL"，"SHALL NOT"，"SHOULD"，"SHOULD NOT"，"RECOMMENDED"，"MAY"和"OPTIONAL"的含意与 RFC2119（参考文档[3]）中描述的相同。

2、配置场景范例（Example deployment scenarios）

下面的场景是一个典型的 Internet 上的 LDAP 目录，并且拥用不同的安全需求。（在下面内容中，"sensitive"（敏感）表示一旦泄露，将导致拥用者受到损失的数据，同时可能存在受到保护但并不敏感（sensitive）的数据）。这里并不试图展示一个全面的列表，而是展示可能的场景，尤其展示在拥有物理保护的网络中的场景。

1、 一个只读目录，它不包含任何敏感数据，允许"anyone"（任何人）访问，并且如果存在 TCP 连接拦截或 IP 窃听也无所谓。该目录除了有管理性服务的限制以外，不需要安全功能。

2、 一个只读目录，它不包含任何敏感数据，但读取访问的赋予（grant）基于身份。并且如果存在 TCP 连接拦截或 IP 窃听也无所谓。这种场景需要一个安全认证功能。

3、 一个只读目录，它不包含任何敏感数据，并且客户端需要去保证目录数据已由服务器验证，同时从服务器返回时不被修改。

4、 一个可读写目录，它不包含任何敏感数据，读访问对"anyone"有效，而更新访问只对经过适当授权的人有效。目前如果存在 TCP 连接拦截也无所谓。这种场景需要一个安全认证功能。

5、 一个包含敏感数据的目录。这种场景需要会话期（session）保密保护**以及**安全认证。

3、认证和授权：定义和概念（Authentication and Authorization: Definitions and Concepts）

本节定义基本术语，概念以及认证、授权、凭证、身份之音的相互关系。这些概念被用于描述多种应用于客户端认证和授权的安全方法。

3.1、访问控制策略（Access Control Policy）

访问控制策略是一个定义资源保护的规则集，通常根据人或其它实体访问那些资源的能力进行设置。一个访问控制策略的通用表示形式是一个访问控制列表（access control list）。安全对象和安全机制（例如：我们在本文档中论讨的）使访问控制策略的表示形式和强制执行成为可能。访问控制策略一般情况下根据下面描述的访问控制属性进行表达。

3.2、访问控制因素（Access Control Factors）

当一个请求被服务器处理时，可能与很多与安全有关的（security-related）因素相关（参考文档[1]的 4.2 节）。该服务器使用这些因素来决定是否以及如何处理该请求。这些因素被称为访问控制因素（ACF）。它可能包括源 IP 地址、加密力度、请求操作的类型、日期时间等。某些因素可能是某种请求所特有的，而某些因素可能与请求传输时经过的连接相关，某些可能是"environmental"（环境方面的）（例如：日期时间）。

访问控制策略根据访问控制因素表达。例如：一个拥有访问控制因素 i,j,k 的请求可以在资源 Z 上执行操作 Y。服务器使这种表达式对其有效的访问控制因素的集合是该服务器实现特定的（implementation-specific）。

3.3、认证，凭证，身份（Authentication, Credentials, Identity）

认证凭证是由一方提供给另一方的证明，它声明了提供该证明一方的身份（例如：一个 user），提供该证明的一方正在试图建立与另一方（一般是一个服务器）的连接。认证是建立传输，并核对这些凭证以及核对该凭证声明的身份的过程。认证身份是在凭证中存在的名称。

存在多种认证凭证的形式——被使用的形式依赖于多方协商的特定的认证机制。例如：X.509 证书，Kerberos 票证（Kerberos ticket），简单身份和密码对儿。应注意，认证机制可以约束与其一起使用的认证身份的形式。

3.4、授权身份（Authorization Identity）

授权身份是一类访问控制因素。它是请求执行一个操作的 user 或其它实体的名称。访问控制策略经常根据授权身份来表达。例如：实体 X 能够在资源 Z 上执行操作 Y。

已绑定到一个关联上的授权身份经常与客户端存在认证身份完全相同，但它们也可能不同。SASL 允许客户端指定一个与来自客户端凭证声明的认证身份不同的授权身份。这样做

就允许代理（例如 proxy 服务器）使用它们自己拥有的凭证进行认证，然后请求它们所代理的认证身份的访问权限（参考文档[2]）。由一个服务（例如：TLS）提供的认证身份形式也可能与用于表达服务器访问控制策略的授权身份不相符，如果这样，就需要执行服务器特定的映射（server-specific mapping）。服务器调整和验证一个来自客户端提供的认证凭证的授权身份的方法是服务器实现特定的（implementation-specific）。

4、需要的安全机制（Required security mechanisms）

面对上述需求，允许任何实现自行选择可能的安全机制并不是一种可能得到互操作性的策略（strategy）。在缺乏权威的情况下，客户端被实现为不支持任何由服务器支持的安全功能，或者更糟的是，客户端仅支持类似明文密码的机制，而这种机制提供了明显不健全的安全。

对于攻击者执行一个攻击，和实现去保护免受攻击两方面来说，主动中间攻击（active intermediary attack）都是最困难的。与没有基于对主动中间攻击的威胁进行适当地预测，就对防范主动中间攻击花大力量相比，仅对恶意客户端和被动窃听攻击（passive eavesdropping attack）进行防护的方法是可取的。

对于一个已配置完的目录来说，如果了解身份所持的分辨名的格式和认证数据在目录中存储位置的机制的需求是很强烈的话，这意味着或者该数据对伪认证（fake authentication）是没有用处的（类似于 Unix 的"/etc/passwd"文件格式），或者该数据的内容从不在未保护线路中传递。也就是说，该数据或者由外部协议更新，或者它仅由对窃听防护良好的会话更新。那么允许认证方法协带基于已存在的 user 身份形式的授权身份，用于提供非 LDAP 认证服务的向后兼容性的需求也是希望的。

所以下列实现应该是符合需求的：

1、 对于一个只读，公开目录来说，可以使用匿名认证（在第 5 节描述）。

2、 提供基于密码认证访问的服务器实现**必须**（MUST）支持使用 DIGEST-MD5 SASL（参考文档[4]）机制的认证（在第 6.1 节描述）。这样做提供了带有防护被动窃听攻击的客户端认证，但不提供防护主动中间攻击。

3、 对于需要会话保护和认证的目录，Start TLS 扩展操作（Start TLS extended operation）（参考文档[5]）以及或者简单认证选择或者 SASL EXTERNAL 机制可以在一起使用。实现**应该**（SHOULD）支持 6.2 节描述的密码认证，并且**应该**（SHOULD）支持 7.1 节描述的证书认证。它们在一起可以为传输中的数据提供完整性和防泄密保护，也可以提供客户端和服务器的认证，包括防护主动中间攻击。

如果 TLS 经协商后，客户端**必须**（MUST）忽略所有在 TLS 协商之前接收自服务器的

信息。特别是，在 TLS 协商后，supportedSASLMechanisms 的值**可以**（MAY）不同（尤其是，EXTERNAL 机制或建议的 PLAIN 机制可能仅在 TLS 协商被执行后才被列出）。

如果 SASL 安全层经过协商，客户商**必须**（MUST）忽略所有在 SAL 协商之前接收自服务器的信息。特别是，如果客户端被配置支持多 SASL 机制（multiple SASL mechanisms），它**应该**（SHOULD）在 SASL 安全层协商的前后都接收 supportedSASLMechanisms，并且在 SASL 安全层被协商后之核对该值未被改变。这样做可以探测某些主动攻击，这些主动攻击可以从 supportedSASLMechanisms 列表中删除所支持 SASL 机制的，并且还允许客户端保证它正在使用由客户端和服务器共同支持的最好的机制（另外，**应该**（SHOULD）允许通过一个不同的凭证源提供给客户端所支持的 SASL 机制列表的环境，例如：作为数字签名对象（digitally signed object）一部分）。

5、匿名认证（Anonymous authentication）

修改条目，或者访问受保护属性或条目的目录操作通常需要客户端认证。不执行任何这些操作的客户端一般使用匿名认证。

LDAP 实现**必须**（MUST）支持在 5.1 节定义的匿名认证。

LDAP 实现**可以**（MAY）支持在 5.2 节定义的带 TLS 的匿名认证。

当防止目录条目的访问的访问控制限制**可能**（MAY）存在时，LDAP 服务器**应该**（SHOULD）允许一个匿名绑定（anonymously-bound）客户端检索根 DSE（root DSE）的 supportedSASLMechanisms 属性。

LDAP 服务器**可以**（MAY）使用其它由底层或扩展方法提供的与客户端相关的信息，来赋予或拒绝甚至是匿名认证客户端的访问。

5.1、匿名认证过程（Anonymous authentication procedure）

没能在一个连接上成功完成绑定操作的 LDAP 客户端是匿名认证的。

LDAP 客户端也**可以**（MAY）在一个绑定请求中指定匿名认证，方法是通过使用一个带有 0 长度八进制字符串的简单认证选项。

5.2、匿名认证和 TLS（Anonymous authentication and TLS）

LDAP 客户端**可以**（MAY）使用 Start TLS 操作（参考文档[5]）来协商 TLS 安全（参考文档[6]）的使用。如果客户端没有预先绑定，则在客户端使用扩展 SASL（EXTERNAL SASL）

机制来协商客户端证书的承认之前，客户端是匿名认证的。

TLS 密码集（ciphersuite）的建议在第 10 节给出。

在 TLS 协商期间请求客户端提供它们的证书的 LDAP 服务器**可以**（MAY）使用一个本地安全策略，在客户端没有提交一个能通过验证的证书的情况下，决定 TLS 协商是否成功完成。

6、基于密码的认证（Password-based authentication）

LDAP 实现**必须**（MUST）支持使用 DIGEST-MD5 SASL 机制对密码进行保护的密码认证，这一内容在 6.1 节定义。

当连接使用 TLS 防护窃听时，LDAP 实现**应该**（SHOULD）支持带有"simple"密码选项的认证。这一内容在 6.2 节定义。

6.1、摘要认证（Digest authentication）

LDAP 客户端**可以**（MAY）通过在根 DSE 上执行一个查询请求来判定服务器是否支持该机制，该查询请求 supportedSASLMechanisms 属性，并且检查是否字符串"DIGEST-MD5"是否作为该属性的值存在。

在认证的第一阶段，当客户端正在执行一个"initial authentication"（初始化认证）（在参考文档[4]的 2.1 节定义）时，客户端发送一个 LDAP 版本号为 3、认证方式为 sasl（sasl 机制名是"DIGEST-MD5"），并且不提交证书的绑定请求。客户端然后等待服务器对该请求的响应。

服务器将返回 resultCode（结果码）为 saslBindInProgress 的绑定响应，并且 serverSaslCreds 域存在于该响应中。该域的内容是由"digest-challenge"（在参考文档[4]的 2.1.1 节定义）定义的字符串。该服务器**应该**（SHOULD）包括域指示（realm indication）并且**必须**（MUST）声明支持 UTF-8。

客户端将发送带有不同消息 ID、LDAP 版本号为 3、认证方式为 sasl（sasl 机制名是"DIGEST-MD5"）、并且凭证包含由"digest-response"（在参考文档[4]的 2.1.2 节定义）定义的字符串的绑定请求。serv-type 是"ldap"。

服务器将返回 resultCode 或者是成功，或者是失败指示的绑定响应。如果认证是成功的，并且服务器不支持进一步的认证，那么 credentials（凭证）域将缺席。如果认证是成功的，并且服务器支持进一步的认证，那么 credentials（凭证）域包括由"response-auth"（在参考文档[4]的 2.1.3 节定义）定义的字符串。在客户端和服务器中对进一步认证的支持是可选的（即，

并不是必须的）。

6.2、TLS 加密下的简单认证选项（"simple" authentication choice under TLS encryption）

拥有包含一个 userPassword 属性的目录条目的 user **可以**（MAY）在提供保密连接（参考文档[6]）的 TLS 加密集协商建立之后，通过执行一个简单密码绑定序列向目录进行认证。

客户端将使用 Start TLS 操作（参考文档[5]）来协商在与 LDAP 服务器的连接上的 TLS 安全（参考文档[6]）的使用。客户端不需要预先已经绑定到目录上。

在该认证过程成功的情况下，客户端和服务器**必须**（MUST）协商一个包含大量强度适当的加密算法的密码集（ciphersuite）。密码集的建议在第 10 节给出。

在 TLS 协商成功完成之后，客户端**必须**（MUST）发送一个 LDAP 版本号为 3、包含 user 的条目的名称的 name 域、和带有一个密码的"simple"认证选项的绑定请求。

服务器将对每个 user 的条目中的每个 userPassword 属性的值，使用区分大小写的匹配来比较客户端提交的密码。如果匹配，服务器将响应 resultCode（结果码）为成功，否则服务器将响应带有 invalidCredentials 的 resultCode。

6.3、TLS 的其它认证选项（Other authentication choices with TLS）

在 TLS 协商之后，执行一个不涉及平文本可重用密码交换的 SASL 认证也是可能的。在这种情况下，如果服务仅需要数据完整性，则客户端和服务器不需要协商一个提供保密的密码集。

7、基于证书的认证（Certificate-based authentication）

LDAP 实现**应该**（SHOULD）支持在 TLS 中利用客户端证书进行认证，这一内容在 7.1 节定义。

7.1、TLS 基于证书的认证（Certificate-based authentication with TLS）

如果服务器请求 user 的证书，一个拥有公钥/私钥（public/private key）对儿，并且公钥已经由一个 CA（Certification Authority）注册的 user 可以使用这个密钥对儿向目录服务器进行认证。User 的证书 subject field（主题域）**应该**（SHOULD）是 user 的目录条目的名字，并且 CA 必须被已经发布了该证书的目录服务器完全信任，目的是为了服务器能够处理该证书。服务器验证证书路径的方法已经超出了本文档所讨论的范围。

服务器**可以**（MAY）支持证书的主题域名与 user 的目录条目的名称不同（译者注：即主题名所描述的 DN 与存放该证书的条目的 DN 不同）时的映射。支持名称映射的服务器**必须**（MUST）具有被配置用来支持不需要映射的证书的能力（译者注：这句话的意思是，目录服务器必须既支持上面所说的证书主题名中的 DN 不证书条目 DN 不同的情况，又要支持主题名 DN 与证书条目 DN 相同的情况，在两个 DN 相同的情况下，是不需要映射的）。

客户端将使用 Start TLS 操作（参考文档[5]）来协商在与 LDAP 服务器的连接上的 TLS 安全（参考文档[6]）的使用。客户端不需要预先绑定到目录上。

在 TLS 协商期间，服务器**必须**（MUST）请求一个证书。客户端将提供它的证书给服务器，并且**必须**（MUST）执行一个基于私钥的加密（private key-based encryption），证明它拥有与该证书相关联的私钥。

当需要在传输过程中保护敏感数据时，客户端和服务器**必须**（MUST）协商一个包含大量强度适当的加密算法的密码集。密码集的建议在第 10 节给出。

服务器**必须**（MUST）核对客户端证书的有效性。服务器通常会检查证书的颁发者（issuer）是一个已知的 CA（译者注：在证书中有一个域称为：issuer 颁发者，该域指出了该证书的颁发者的 DN。Issuer 其实也是一张证书，只不过这张证书是 CA 自己所拥有的，一般被称为根证，所有的证书全部由根证颁发，而根证是由自己颁发给自己的，所以根证的 subject 主题和 issuer 颁发者是一样的。根证也要位于目录当中。），并且在客户端的证书链中没有任何证书是无效的或被撤消的。服务器执行这些检查的方法有很多种。

在 TLS 协商成功完成后，客户端将发送一个带有 SASL"EXTERNAL"机制的 LDAP 绑定请求。

8、其它机制（Other mechanisms）

LDAP"simple"认证选项并不能满足 Internet 上的认证需要，在 Internet 上没有网络或传输层保密。

当 LDAP 包括本地（native）匿名和平文本认证方法时，"ANONYMOUS"和"PLAIN"的 SASL 机制不能与 LDAP 一起使用。如果客户端请求了一个与 DN 格式不同的授权身份，那么一个在传输中保护密码的机制**应该**（SHOULD）被使用。

下列基于 SASL 的机制并不在本文档的考虑范围之内：KERBEROS_V4，GSSAPI 和 SKEY。

"EXTERNAL"的 SASL 机制可以被用于请求 LDAP 服务器在一个更底层中使用安全凭证交换。如果 TLS 会话在 SASL EXTERNAL 绑定请求之前未在客户端和服务器之间被建立，并且没有其它扩展认证凭证源（例如：IP 层安全（参考文档[8]）），或者在 TLS 会话建立过程中，服务器没有请求客户端的认证凭证，则 SASL EXTERNAL 绑定**必须**（MUST）失败，结果码为 inappropriateAuthentication。任何客户端认证和与 LDAP 关联的授权状态全部丢失，因此在失败之后，LDAP 关联是一个匿名状态。

9、授权身份（Authorization Identity）

授权身份由 LDAP 绑定请求和响应中的 SASL credentials 域的一部分携带。

当"EXTERNAL"机制正在被协商时，如果 credentials 域存在的话，它包含一个下面描述的 authzId 格式的授权身份。

其它机制定义授权身份在 credentials 域中的位置。

授权身份是一个 UTF-8 字符集字符串，与下面的 ABNF（参考文档[7]）一致：

```
; Specific predefined authorization (authz) id schemes are
; defined below -- new schemes may be defined in the future.

authzId       = dnAuthzId / uAuthzId

; distinguished-name-based authz id.
dnAuthzId    = "dn:" dn
dn               = utf8string       ; with syntax defined in RFC 2253

; unspecified userid, UTF-8 encoded.
uAuthzId     = "u:" userid
userid         = utf8string       ; syntax unspecified
```

一个 UTF8 字符串被定义为一个或多个 ISO10646 字符的 UTF-8 编码。

所有支持认证凭证（例如：密码或证书）存储在目录中的服务器**必须**（MUST）支持 dnAuthzId 选项。

uAuthzId 选项为那些希望在本地目录进行认证，但不知道它们所拥有的 DN 或不知道是否拥有一个目录条目的客户端应用程序提供兼容性。该字符串的格式被定义为仅是 ISO10646 字符的 UTF-8 编码序列，并且进一步解释为在客户端和服务器之间的预先协定。

例如：userid 可以标识一个特定目录服务的 user，该 user 可能是一个登录名也可能是 RFC822 email 地址的本地部分（local-part of an RFC 822 email address）。总的来说，uAuthzId **绝不**（MUST NOT）能被认为是全局唯一的。

另外的授权身份方案（scheme）**可能**（MAY）在本文档的未来版本中定义。

10、TLS 密码集（TLS Ciphersuites）

下列密码集（在参考文档[6]中定义）**绝不**（MUST NOT）能用于密码和数据的加密保护：

TLS_NULL_WITH_NULL_NULL
TLS_RSA_WITH_NULL_MD5
TLS_RSA_WITH_NULL_SHA

下列密码集（在参考文档[6]中定义）能被很容易的破解（在 1997 年，在一个标准 CPU 上使用少于一周的 CPU 时间）。客户端和服务器**应该**（SHOULD）在使用这些密码集之间，谨慎在考虑接受保护的密码和数据：

TLS_RSA_EXPORT_WITH_RC4_40_MD5
TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
TLS_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_DSS_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_anon_EXPORT_WITH_RC4_40_MD5
TLS_DH_anon_EXPORT_WITH_DES40_CBC_SHA

下列密码集易到人工介入（man-in-the-middle）攻击，并且**不应该**（SHOULD NOT）用于保护密码和敏感数据，除非网络配置可以防止人工介入攻击的危险：

TLS_DH_anon_EXPORT_WITH_RC4_40_MD5
TLS_DH_anon_WITH_RC4_128_MD5
TLS_DH_anon_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_anon_WITH_DES_CBC_SHA
TLS_DH_anon_WITH_3DES_EDE_CBC_SHA

支持 TLS 的客户端或服务器**必须**（MUST）至少支持

TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA

11、LDAP 的 SASL 服务名（SASL service name for LDAP）

为了与 SASL（参考文档[2]）一起使用，某个协议必须指定一个服务名以便与不同的 SASL 机制一起使用，例如 GSSAPI。对于 LDAP，服务名为"ldap"，这个名已经在 IANA 注册，作为一个 GSSAPI 服务名。

12、安全考虑（Security Considerations）

安全问题贯穿于本文档，结论是强制的安全是重要的，并且在考虑防范窃听时，会话加密是必要的。

鼓励服务器防止匿名 user 的修改操作。服务器也可以希望通过使用空闭连接超时来最小化服务攻击，并且返回 unwillingToPerform 结果码，而不执行由未授权客户端请求的代价高昂的操作。

客户端未执行 Start TLS 操作的，或者没有为连接完整性和加密服务协商一个适当 SASL 机制的连接易受到人工介入（man-in-the-middle）攻击，该攻击可能在传输中查看和修改信息。

另外的与 EXTERNAL 机制去协商 TLS 相关的安全考虑可以在参考文档[2]，[5]和[6]中找到。

13、感谢（Acknowledgements）

本文档是 IETF 的 LDAPEXT 工作组的成果。该工作组成员的贡献是非常值得赞赏的。

14、参考书目（References）

[1] Wahl, M., Howes, T. and S. Kille, "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997.

[2] Myers, J., "Simple Authentication and Security Layer (SASL)", RFC 2222, October 1997.

[3] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[4] Leach, P. and C. Newman, "Using Digest Authentication as a SASL Mechanism", RFC 2831, May 2000.

[5] Hodges, J., Morgan, R. and M. Wahl, "Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security", RFC 2830, May 2000.

[6] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[7] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[8] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

15、作者地址（Authors' Addresses）

Mark Wahl
Sun Microsystems, Inc.
8911 Capital of Texas Hwy #4140
Austin TX 78759
USA
EMail: M.Wahl@innosoft.com

Harald Tveit Alvestrand
EDB Maxware
Pirsenteret
N-7462 Trondheim, Norway
Phone: +47 73 54 57 97
EMail: Harald@Alvestrand.no

Jeff Hodges
Oblix, Inc.
18922 Forge Drive
Cupertino, CA 95014
USA
Phone: +1-408-861-6656
EMail: JHodges@oblix.com

RL "Bob" Morgan
Computing and Communications
University of Washington
Seattle, WA 98105
USA

Phone: +1-206-221-3307
EMail: rlmorgan@washington.edu

16、完整的版权声明（Full Copyright Statement）

略