表F-3 （来自RFC 2617的）摘要Authentication-Info首部指令

指　令	描　述
nextnonce	nextnonce 指令的值是服务器希望客户端为未来的认证响应使用的 nonce。服务器可能会发送带有 nextnonce 字段的 Authentication-Info 首部，作为实现一次性 nonce 或修改 nonce 的手段。如果提供了 nextnonce 字段，客户端在为下一条请求构建 Authorization 首部时就应该使用它。客户端如果没能做到，就会收到来自服务器的 "stale=TRUE" 认证请求。 服务器实现应该仔细地考虑使用这种机制带来的性能影响。如果每条响应都包含了必须在服务器接收的下一条请求中使用的 nextnounce 指令，就不可能使用管道化请求了。应该考虑在性能和安全之间进行一些平衡，允许在有限的时间内使用老的 nonce 值，以实现请求的管道化。使用 nounce 计数可以在不影响管道化的情况下，维护一个新的服务器 nonce 的大部分安全优势
qop	说明了服务器应用到响应上的“安全保障”选项。值 auth 说明要进行认证，值 auth-int 说明要进行带有完整性保护的认证。服务器在响应中使用的 qop 指令值应该与客户端在相应请求中发送的值相同

rspauth	response auth 指令中的可选响应摘要支持双向认证——服务器证明了它知道用户的密码，而且通过 qop="auth-int"，它还为响应提供了有限的完整性保护。除了当 qop="auth" 或者没有在 Authorization 首部为请求指定 qop 的情况，response-digest 值的计算方式与 Authorization 首部的 request-digest 类似，A2 为：
	`A2 = ":" digest-uri-value`
	当 qop="auth-int" 时，A2 为：
	`A2 = ":" digest-uri-value ":" H(entity-body)`
	其中 digest-uri-value 是请求的 Authorization 首部中 uri 指令的值。cnonce 和 nc 值一定要与此报文所响应的客户端请求中的相应值相同。如果指定了 qop="auth " 或者 qop="auth-int"，就必须提供 rspauth 指令

cnonce	cnonce 值一定要与此报文所响应的客户端请求中的相应值一样。如果指定了 qop="auth " 或 qop="auth-int"，就必须提供 cnonce 指令
nc	nc 值一定要与此报文所响应的客户端请求中的相应值一样。如果指定了 qop="auth " 或 qop="auth-int"，就必须提供 nc 指令
<extension>	未来可以通过这条指令进行扩展。所有不识别的指令都要忽略掉

F.4 参考代码

下列代码实现了 RFC 2617 中 H(A1)、H(A2)、request-digest 和 response-digest 的计算。它使用了 RFC 1321 中的 MD5 实现。