RFC 3261：22章

3.13 HTTPダイジェスト認証

SIPは、HTTPなどと同じく、リクエストメッセージを送出することでサーバ（UASやプロキシサーバ）に対して処理要求を行うリクエスト−レスポンスタイプのプロトコルです。このようなプロトコルの場合、リクエストを受信するサーバ側で「リクエストを送信したクライアントは誰なのか」ということを知ることが必要となる場合があります。SIPでは、サーバ側でリクエストの送信者が誰かということを確認する手段として、HTTPダイジェスト認証方式とよばれる認証方式の利用が規定されています。

3.13.1 チャレンジ−レスポンス方式

SIPでは、リクエストの認証メカニズムとして**HTTPダイジェスト認証**と呼ばれる**チャレンジ−レスポンス方式**の認証機構を規定しています。チャレンジ−レスポンス方式のおおまかな認証手順は次の通りになります。

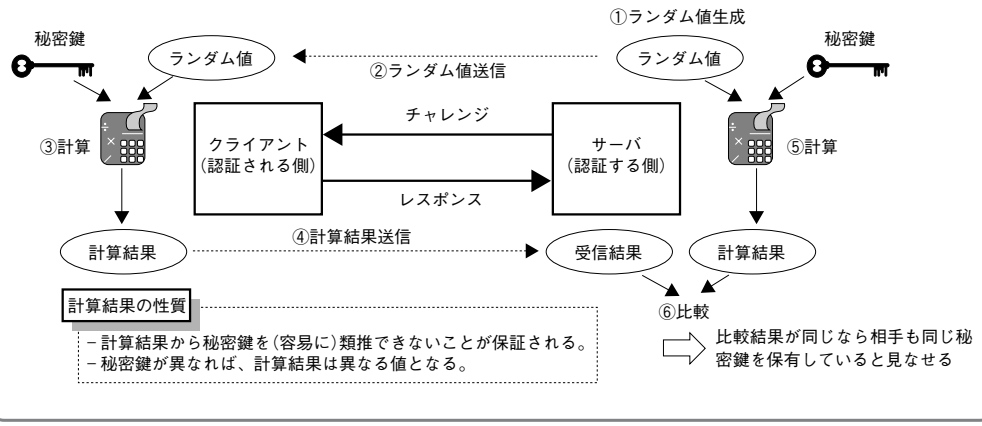
1. 認証を行うサーバがランダムな値を生成する（図3−72①）。
2. 生成したランダム値を認証されるクライアントに送信（チャレンジ）する（②）。
3. クライアント側ではサーバから受信したランダムな値と、双方で共有するパスワードなどの値（秘密鍵）とを組み合わせて計算し（③）、その結果をサーバ側へ送信（レスポンス）する（④）。
4. サーバ側では、計算結果がローカルに計算（⑤）した値と合致した場合に、相手が秘密鍵を知っているものとみなして認証成功とする（⑥）。

チャレンジ−レスポンス方式の利点は、秘密鍵がネットワークにそのままの形で送信されることがなく、仮にメッセージが盗聴された場合でも、秘密鍵の情報が盗聴を行う第三者に漏洩することがない点にあります。

なお、本節の記述においては、チャレンジ−レスポンス方式で用いられる認証データの計算結果である値を`response`値（“`response`”フィールドの値の場合）もしくは〈レスポンス値*139〉（“`response`”以外のフィールドを含む場合）と表記し、SIPメッセージのレスポンスと区別することとします。

＊139 〈レスポンス値〉のことを信用情報（credential）と呼ぶこともある。

図3−72　チャレンジ-レスポンス方式の概要



3.13.2　HTTP ダイジェスト認証方式

リクエストを認証するためのチャレンジ−レスポンス方式の認証規定は、HTTP のリクエスト認証のために策定された**HTTP ダイジェスト認証方式**＊140 を基本的には採用しています。HTTP ダイジェスト認証の SIP への流用は、必要な機能をモジュール化し転用する、というインターネットのプロトコルの利点を活かした典型的な例の1つと言えるでしょう。

HTTP ダイジェスト認証方式では、認証情報を要求するサーバ（UAS）からのメッセージに次の形式の**WWW-Authenticate** ヘッダを含めて送信します。

WWW-Authenticate: Digest〈チャレンジ値〉

〈チャレンジ値〉の部分には、認証において必要な要素を、“パラメータ名”＝“パラメータ値”のフォーマットでカンマにより区切って並べたものが入ります。

これに対して、認証される側のクライアントからは、受信した〈チャレンジ値〉の内容を元に〈レスポンス値〉を算出し、送出するメッセージに次の形式の**Authorization** ヘッダを含めて送信します。〈レスポンス値〉の算出にはユーザ名とパスワードの値が使われており、パスワードを知らなければ算出できないような計算が行われます。

Authorization: Digest〈レスポンス値〉

＊140 HTTP ダイジェスト認証方式の詳細な動作規定は RFC 2617 に記述されており、RFC 3261 には、SIP での適用方法と、SIP で利用する場合の機能的な差分についてのみが規定されている。

Authorizationヘッダを受信したサーバ側では〈レスポンス値〉の値の正当性を判定し、ユーザの認証を行います。

受信した〈レスポンス値〉に含まれるユーザ名やその他の情報およびサーバで保持しているユーザ名に対応するパスワードを元にサーバ側でもクライアント側と同じ計算を行います。この計算結果と受信した値が一致した場合に、クライアントがパスワードを知っている正当なユーザである、とサーバ側で認証することができます。

なお、UASではなく、プロキシサーバがUACの認証を行う場合には、**Proxy-Authenticate**ヘッダと**Proxy-Authorization**ヘッダがそれぞれ**WWW-Authenticate**ヘッダと**Authorization**ヘッダの代わりに用いられます。

●HTTPダイジェスト認証方式のパラメータ詳細

表3−37にHTTPダイジェスト認証の〈チャレンジ値〉に使用されるパラメータを、表3−38に〈レスポンス値〉に使用されるパラメータをそれぞれ示します。また、SIPにおけるチャレンジ-レスポンス（**WWW-Authenticate**ヘッダと**Authorization**ヘッダ）の例を図3−73に示します。

HTTPダイジェスト認証の大きな特長の1つは、サーバ側でチャレンジ送出後にチャレンジに関する状態を保持することなく動作が可能であるという点にあります。**response**値の算出に必要な情報は、全てクライアント側から送信される情報に含まれています（サーバから送出した**nonce**や**realm**なども全てコピーされて返送されます）。したがって、サーバ側では、認証のためのエラーレスポンス（**401**（**Unauthorized**）レスポンスや**407**（**Proxy Authentication Required**）レスポンス）の内容や元となるリクエストに関する個別情報を保持している必要がありません。この状態を持たないという性質（ステートレス）は、DoS攻撃の耐性の面からも非常に重要です。認証されないリクエストにより、一定時間システム上で何らかのリソースの使用が発生する（ステートが発生する）とすれば、DoS攻撃に対する脆弱性の原因となります。サーバ側でステートを保持しないことで、認証されていないメッセージによる不要なリソースの使用を最小化することができ、DoS攻撃への耐性を高めることが可能となります。

3.13.3 UAS-UAC間の認証

UASがHTTPダイジェスト認証方式を利用してUACを認証する場合には、まずUASは**401**（**Unauthorized**）レスポンスを返送します。**401**（**Unauthorized**）レスポンスにはUASからの〈チャレンジ値〉を含む**WWW-Authenticate**ヘッダを含めます。

このレスポンスを受信したUACは、その内容に従い〈レスポンス値〉を生成し、**Authorization**ヘッダに含めてリクエストを再送信します。このときには、**401**（**Unauthorized**）レスポンスを受信することとなったリクエストと同じ**Call-ID**ヘッダの値を

表3−37　SIPにおけるHTTPダイジェスト認証の〈チャレンジ値〉に含まれるパラメータ

パラメータ	M/O	値	用　途
realm	M	ホスト名/ドメイン名	認証を要求するサーバのホスト名/ドメイン名を示す。利用するユーザ名/パスワードの組みを決定するためのキーとして利用されるとともに、認証情報の入力を促す場合にユーザへの表示にも使われることもある。
domain	O	URI	認証が有効なスペース（protection space）のURIのリストを指定する。ただし、SIPではprotection spaceの概念が不要なため、有用な情報ではない。
nonce	M	base64か16進数文字列	サーバが生成するランダム値。サーバはnonceを時間情報と関連付けた値にすることにより、生成後一定の期間だけ有効とすることも可能。
opaque	O	base64か16進数文字列	サーバが設定する任意の文字列。この値は、クライアントで認証レスポンスにコピーされて返送されるため、サーバ側では、この値を利用して必要な処理（負荷分散複数サーバの振分け等）が可能となる。
stale	O	TRUEまたはFALSE（大文字/小文字の区別なし）	クライアントからのresponse値が有効でなかった場合に返送するエラーレスポンスに含まれる。サーバが受信したresponse値の計算結果は正しいが、nonce値が有効でない場合には“TRUE”が設定される。response値の計算結果が異なる場合（パスワードが違う）や指定されたユーザ名が存在しないなどの場合には、“FALSE”が設定されるか、省略される。
algorithm	O	“MD5”，“MD5-sess”などのカンマ区切りリスト（大文字/小文字の区別なし）	response値を計算するときに利用可能なアルゴリズムのリストを示す。HTTPダイジェスト認証方式では、“MD5”と呼ばれるハッシュ値計算方式がデフォルトで、algorithmパラメータが含まれていなかった場合には“MD5”が利用される。
qop	O	“auth”，“auth-int”などのカンマ区切りリスト（大文字/小文字の区別なし）	“quality of protection”の略。サポートする認証の強化方式を指定する。qopパラメータを含めた場合には、cnonceやncなどのセキュリティを強化するための機能を利用することを示す＊141。

M：必須（Mandatory），O：オプション（Option）

利用し、CSeqヘッダの値を1つだけ増加させます。

ただし、401（Unauthorized）レスポンスを受信することとなったリクエストにUACが既に信用情報を含めていて、その結果受信した〈チャレンジ値〉のstaleの値がFALSEとなっていた場合には、ユーザ名やパスワードが異なっていたということを意味しており、同じユーザ名とパスワードでリクエストを自動的に再送信することは避ける必要があります。

＊141 RFC 2617ではオプションだが、RFC 3261では、qopパラメータを含めることを必須としている。現時点では、全てのクライアントがqopパラメータに対応する機能を実装しているわけではない。