

## 2.2.7.16 LSAPR\_TRUSTED\_DOMAIN\_AUTH\_BLOB

The **LSAPR\_TRUSTED\_DOMAIN\_AUTH\_BLOB** structure contains a counted buffer of authentication material. Domain trust authentication is specified in [\[MS-ADTS\]](#) section 7.1.6.9.1.

```
typedef struct _LSAPR_TRUSTED_DOMAIN_AUTH_BLOB {
    unsigned long AuthSize;
    [size_is(AuthSize)] unsigned char* AuthBlob;
} LSAPR_TRUSTED_DOMAIN_AUTH_BLOB,
*PLSAPR_TRUSTED_DOMAIN_AUTH_BLOB;
```

**AuthSize:** The count of bytes in **AuthBlob**. [<24>](#)

**AuthBlob:** An array of bytes containing the authentication material. If the **AuthSize** field has a value other than 0, this field MUST NOT be NULL. Always encrypted using algorithms, as specified in section [5.1.1](#). The plaintext layout is in the following format.

The incoming and outgoing authentication information buffer size included at the end of the LSAPR\_TRUSTED\_DOMAIN\_AUTH\_BLOB can be used to extract the incoming and outgoing authentication information buffers from the LSAPR\_TRUSTED\_DOMAIN\_AUTH\_BLOB. Each of these buffers contains the byte offset to both the current and the previous authentication information. This information can be used to extract current and (if any) previous authentication information.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
512 bytes of random data ...																															
CountOutgoingAuthInfos																															
ByteOffsetCurrentOutgoingAuthInfo																															
ByteOffsetPreviousOutgoingAuthInfo																															
CurrentOutgoingAuthInfos																															
...																															
PreviousOutgoingAuthInfos (optional)																															
...																															
CountIncomingAuthInfos																															
ByteOffsetCurrentIncomingAuthInfo																															
ByteOffsetPreviousIncomingAuthInfo																															

CurrentIncomingAuthInfos
...
PreviousIncomingAuthInfos (optional)
...
OutgoingAuthInfoSize
IncomingAuthInfoSize

**CountOutgoingAuthInfos:** Specifies the count of entries present in the **CurrentOutgoingAuthInfos** field.

**ByteOffsetCurrentOutgoingAuthInfo:** Specifies the byte offset from the beginning of **CountOutgoingAuthInfos** to the start of the **CurrentOutgoingAuthInfos** field.

**ByteOffsetPreviousOutgoingAuthInfo:** Specifies the byte offset from the beginning of **CountOutgoingAuthInfos** to the start of the **PreviousOutgoingAuthInfos** field. If the difference between **ByteOffsetPreviousOutgoingAuthInfo** and **OutgoingAuthInfoSize** is 0, this field MUST be ignored; this also means that the **PreviousOutgoingAuthInfos** field has zero entries.

**CurrentOutgoingAuthInfos:** Contains an array of **CountOutgoingAuthInfos** of [LSAPR\\_AUTH\\_INFORMATION \(section 2.2.7.17\)](#) entries in self-relative format.

**PreviousOutgoingAuthInfos:** Contains an array of **CountOutgoingAuthInfos** [LSAPR\\_AUTH\\_INFORMATION](#) entries in self-relative format. See the comments for the **ByteOffsetPreviousOutgoingAuthInfo** field to determine when this field is present.

**CountIncomingAuthInfos:** Specifies the count of entries present in the **CountIncomingAuthInfos** field.

**ByteOffsetCurrentIncomingAuthInfo:** Specifies the byte offset from the beginning of **CountIncomingAuthInfos** to the start of the **CurrentIncomingAuthInfos** field.

**ByteOffsetPreviousIncomingAuthInfo:** Specifies the byte offset from the beginning of **CountIncomingAuthInfos** to the start of the **PreviousIncomingAuthInfos** field. If the difference between **ByteOffsetPreviousIncomingAuthInfo** and **IncomingAuthInfoSize** is 0, this field MUST be ignored; this also means that the **PreviousIncomingAuthInfos** field has zero entries.

**CurrentIncomingAuthInfos:** Contains an array of **CountIncomingAuthInfos** [LSAPR\\_AUTH\\_INFORMATION](#) entries in self-relative format.

**PreviousIncomingAuthInfos:** Contains an array of **CountIncomingAuthInfos** [LSAPR\\_AUTH\\_INFORMATION](#) entries in self-relative format. See the comments for the **ByteOffsetPreviousIncomingAuthInfo** field to determine when this field is present.

**OutgoingAuthInfoSize:** Specifies the size, in bytes, of the subportion of the structure from the beginning of the **CountOutgoingAuthInfos** field through the end of the of the **PreviousOutgoingAuthInfos** field.

**IncomingAuthInfoSize:** Specifies the size, in bytes, of the sub-portion of the structure from the beginning of the **CountIncomingAuthInfos** field through the end of the of the **PreviousIncomingAuthInfos** field.

### 2.2.7.17 LSAPR\_AUTH\_INFORMATION

The **LSAPR\_AUTH\_INFORMATION** structure communicates information about authentication between trusted domains. Domain trust authentication is specified in [\[MS-ADTS\]](#) section 7.1.6.9.1.

```
typedef struct _LSAPR_AUTH_INFORMATION {
    LARGE_INTEGER LastUpdateTime;
    unsigned long AuthType;
    unsigned long AuthInfoLength;
    [size_is(AuthInfoLength)] unsigned char* AuthInfo;
} LSAPR_AUTH_INFORMATION,
*PLSAPR_AUTH_INFORMATION;
```

**LastUpdateTime:** The date and time when this authentication information was last updated. It is a 64-bit value that represents the number of 100-nanosecond intervals since January 1, 1601, UTC.

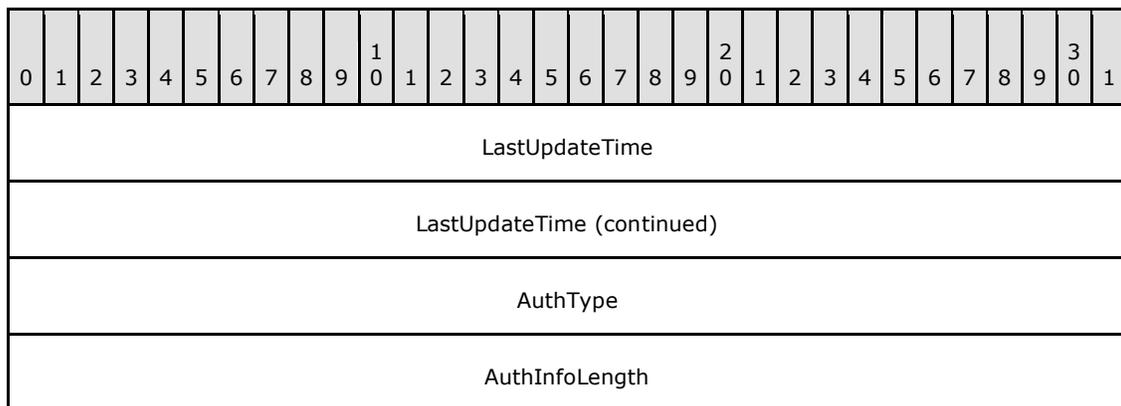
**AuthType:** A type for the AuthInfo, as specified in the following table.

Value	Meaning
0x00000000	This type MUST be ignored.
0x00000001	Derived RC4HMAC key. For more information, see <a href="#">[RFC4757]</a> .
0x00000002	A plaintext password. Indicates that the information stored in the attribute is a Unicode plaintext password. If this AuthType is present, Kerberos can then use this password to derive additional key types that are needed to encrypt and decrypt cross-realm TGTs.
0x00000003	A plaintext password version number.

**AuthInfoLength:** The count of bytes in AuthInfo buffer. [<25>](#)

**AuthInfo:** Authentication data that depends on the **AuthType**.

The self-relative form of the **LSAPR\_AUTH\_INFORMATION** structure is used in [LSAPR\\_TRUSTED\\_DOMAIN\\_AUTH\\_BLOB](#); in that case, the structure memory layout looks like the following.



AuthInfo [1 ... AuthInfoLength]