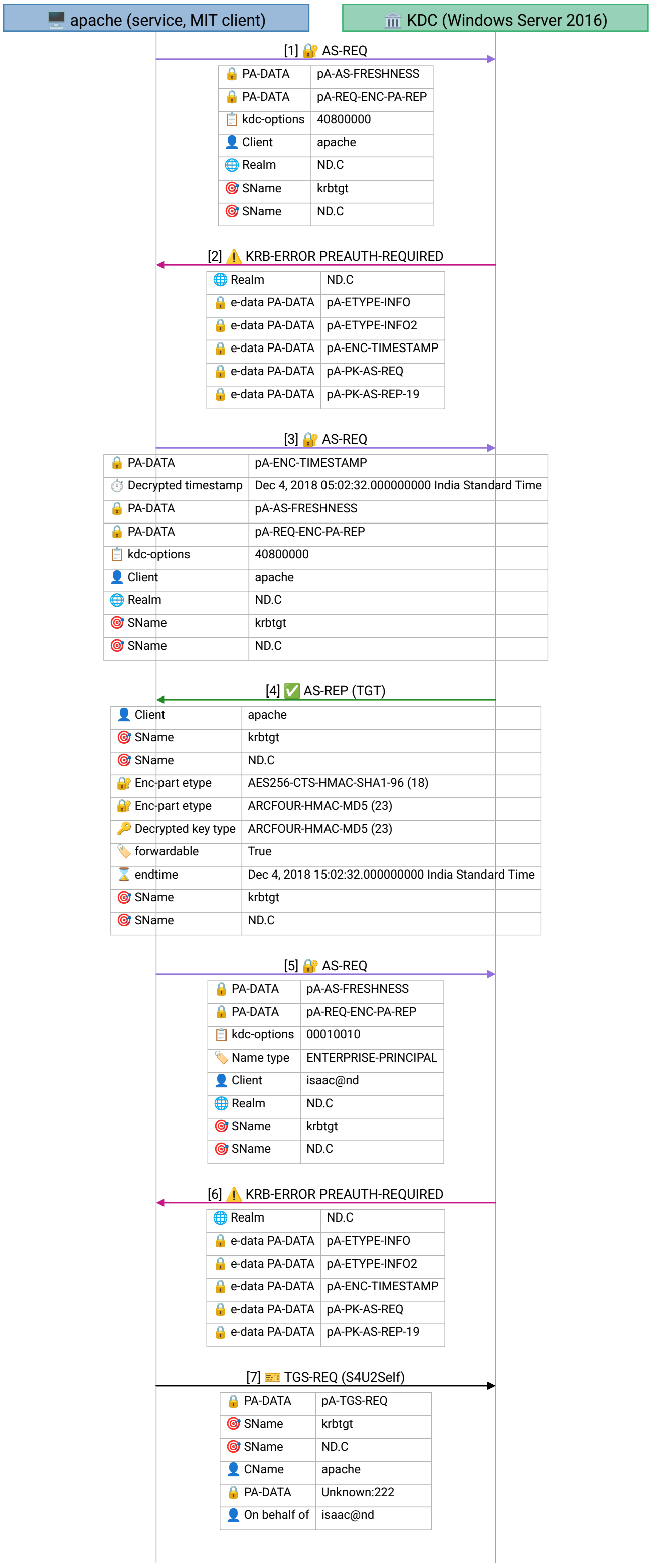
kerberos-s4u2self-protocol-transition.annotated



[Frame 1] Phase 1: the service gets its own TGT. An MIT Kerberos client, running as the service account `apache` in realm `ND.C`, asks a Windows Server 2016 KDC for a TGT. It offers only RC4 and sends no pre-authentication yet.

`PA-AS-FRESHNESS` and `PA-REQ-ENC-PA-REP` are empty markers: the client says it supports freshness tokens (RFC 8070) and wants the reply's padata protected (RFC 6806).

[Frame 2] `PREAUTH_REQUIRED` (error 25): the KDC wants proof of the password first. The error lists the methods it accepts, encrypted timestamp and PKINIT, and, in `ETYPE-INFO2`, the key type to use: RC4. RC4 keys have no salt, so none is listed.

[Frame 3] The retry adds `PA-ENC-TIMESTAMP`: the current time encrypted with apache's key. The keytab holds that key, so we can read it: 4 Dec 2018 05:02:32.

[Frame 4] The AS-REP carries apache's TGT, encrypted with the `krbtgt` key as AES-256 (the KDC's choice for its own ticket; the keytab can't open it). The enc-part, under apache's RC4 key, holds the TGT session key and the flags `forwardable, renewable, initial, pre-authent`, valid for 10 hours.

[Frame 5] Phase 2: find the user's realm. The service now acts for user `isaac`, who has not authenticated to it with Kerberos. The user might have signed in to the website with a form or a certificate instead.

First it sends an AS-REQ in isaac's name, as an enterprise principal `isaac@nd`, with no pre-authentication. It cannot log in as isaac and doesn't try. MS-SFU 3.1.5.1.1.2 describes this lookup: the KDC that holds the account answers `PREAUTH_REQUIRED`, and any other KDC answers with a referral.

[Frame 6] `PREAUTH_REQUIRED`: isaac's account is in this KDC's realm, `ND.C`. The lookup is done.

[Frame 7] Phase 3: S4U2Self. The service asks for a ticket to itself (`sname` = `apache`), presenting its own TGT in `PA-TGS-REQ`, but on behalf of isaac:

- `PA-FOR-USER` (129) names `isaac@nd`, with a checksum keyed by the TGT session key so the KDC knows the service, not someone on the path, chose the name (MS-SFU 2.2.1),
- `PA-S4U-X509-USER` (130) names the user again, the newer form of the same request (MS-SFU 2.2.2),
- `kdc-options` asks for `forwardable`.

🖥️ apache (service, MIT client) | 🏛️ KDC (Windows Server 2016)

🔒 PA-DATA	pA-FOR-X509-USER
👤 On behalf of	isaac@nd
📋 kdc-options	40810000
🎯 SName	apache

[8] ✅ TGS-REP

🔒 PA-DATA	pA-FOR-X509-USER
👤 Client	isaac@nd
🌐 Realm	ND.C
🎯 SName	apache
🏷️ forwardable	False
🔑 Decrypted key type	ARCFOUR-HMAC-MD5 (23)
👤 Client	isaac@nd
🪪 PAC name	isaac@nd
🔑 Decrypted key type	ARCFOUR-HMAC-MD5 (23)
🏷️ forwardable	False
🎯 SName	apache
🔒 PA-DATA	pA-SUPPORTED-ETYPES
🔒 PA-DATA	pA-FOR-X509-USER

There is also a padata type 222 that Wireshark does not recognize.

[Frame 8] A ticket to apache in isaac's name, without isaac. Opened with apache's key from the keytab:

- client `isaac@nd`, service `apache`,
- a PAC for isaac: user RID 1103, group 513 (Domain Users), UPN `isaac@nd.c`, and logon count 0: the account has never logged on at this domain controller,
- flags `renewable`, `pre-authent`, but **not** `forwardable`, although the request asked for it.

MS-SFU 3.2.5.1.2 ties `forwardable` on an S4U2Self ticket to whether the service is trusted to authenticate for delegation, and this KDC left it off, so apache is evidently not trusted for it. The service can use this ticket to learn isaac's groups and authorize the user locally, but it can't pass it on through classic constrained delegation. Resource-based delegation can still use it, as the RBCD capture shows.

Because the authenticator in frame 7 carried a subkey, the KDC encrypted the reply's enc-part with that subkey rather than the TGT session key (RFC 4120 5.4.2).