KERBEROSv5

Client

Unknown

Password

x509 Certificate

signedAuthPack

PKINIT

eTYPE-ARCFOUR-HMAC-MD5 (23)

User Secret Key = MD4 (UTF-16LE (<Password>))

eTYPE-AES256-CTS-HMAC-SHA1-96 (18)

Salt: Machine Account: Upper (<Domain>) + "host" + Lower (<SamAccountName>[:-1]) + "." + Lower (<Domain>)

Salt: User Account: Upper (<Domain>) + <SamAccountName>

User Secret Key = string_to_key (UTF8 (<Password>), <Salt>, None, Seedsize = 32, BlockSize = 32, EncType = 18)

eTYPE-AES128-CTS-HMAC-SHA1-96 (17)

Salt: Machine Account: Upper (<Domain>) + "host" + Lower (<SamAccountName>[:-1]) + "." + Lower (<Domain>)

Salt: User Account: Upper (<Domain>) + <SamAccountName>

User Secret Key = string_to_key (UTF8 (<Password>), <Salt>, None, Seedsize = 16, BlockSize = 16, EncType = 17)

User Secret Key

KRB_AS_REQ

PADATA	First Request	PA-PAC-REQUEST (128)
	Second Request	PA-PAC-REQUEST (128) + PA-ENC-TIMESTAMP (2)
		PA-ENC-TIMESTAMP (2) Cipher = Encryption (<EncType>, < User Secret Key>, 1, <Confounder>, EncodeData (<DateStr_UTC>))
	PKINIT Authentication	PA-PAC-REQUEST (128) + PA-PK-AS-REQ (16)
		PA-PK-AS-REQ (16) Value = EncodeData (<PFXFileData>, < CuSec>, <CTime>, <Nonce>, <PaChecksum>, < DHParams>, <DHPrivKey>, <DHNonce>)
REQ-BODY	kdc-options = Flags requested for the resulting ticket	
	cname = Client name	
	realm = Domain name	
	sname = Requested service = krbtgt/<Domain>	
	till = Requested expiration time of the requested TGT	
	rtime = Requested renewal expiration time if renewable ticket was requested	
	nonce = Randomly generated message nonce	
	etype = Requested encryption types	
	addresses = Client IP address	

Key Distribution Center (KDC)

KDC_ERR_C_PRINCIPAL_UNKNOWN

KDC_ERR_PREAUTH_REQUIRED

<Unknown>

Parse KRB_AS_REQ

<cname> not found in database

Sent only PA-PAC-REQUEST (128) and <cname> require Pre-Auth

PA-ENC-TIMESTAMP (2) contain <Timestamp> not in 5 minutes (default) delay

PA-PK-AS-REQ (16) Compute <AS-Rep Encryption key>

Send KRB_AS_REP

KRB_AS_REP

PADATA	Basic Authentication	PA-ETYPE-INFO2 (19)
	PKINIT Authentication	PA-PK-AS-REP (17)
crealm		
cname		
Ticket Granting Ticket = TGT	tkt-vno = Ticket format version number	
	realm	
	sname = krbtgt/<Domain>	
	etype	
		Flags = Ticket flags
		Session Key = <Client-to-TGS Session Key>
		crealm
		cname
		transited
		authtime
		Optional - starttime
		endtime
		Optional - renew-till
		Optional - caddr
		LogonTime
		LogoffTime
		KickOffTime
		PasswordLastSet
		PasswordCanChange
		EffectiveName
		FullName
		LogonScript
		ProfilePath
		HomeDirectory
		HomeDirectoryDrive
		LogonCount
		BadPasswordCount
		UserId
		PrimaryGroupId
		GroupCount
	Optional - PAC_LOGON_INFO	GroupIds
		UserFlags
		UserSessionKey
		LogonServer
		LogonDomainName
		LogonDomainId
		LMKey
		UserAccountControl
		SubAuthStatus
		LastSuccessfulILogon
	<TicketPart> encrypted with <TGS Secret Key> = <Krbtgt Secret Key>	LastFailedILogon
		FailedILogonCount
		SidCount
		ExtraSids
		ResourceGroupDomainSid
	Optional - authorization-data	ResourceGroupCount
		ResourceGroupIds
		ClientId
	Optional - PAC_CLIENT_INFO_TYPE	NameLength
		Name
		UpnLength
		UpnOffset
		UpnName
		DnsDomainNameLength
		DnsDomainNameOffset
		DnsName
	Optional - PAC_UPN_DNS_INFO	Flags
		Optional - SamNameLength
		Optional - SamNameOffset
		Optional - SamName
		Optional - SidLength
		Optional - SidOffset
		Optional - Sid
	Optional - PAC_SERVER_CHECKSUM	SignatureType
		Signature
	Optional - PAC_PRIVSVR_CHECKSUM	SignatureType
		Signature
		Version
		EncryptionType
	PKINIT Authentication - PAC_CREDENTIALS_INFO	CredentialCount
	SerializedData encrypted with <AS-Rep Encryption Key>	Version
		Credentials: Flags
		Credentials: LmPassword
		Credentials: NtPassword
	Optional - PAC_DELEGATION_INFO	
	Optional - PAC_ATTRIBUTES_INFO	
	Optional - PAC_REQUESTOR_INFO	
	Session Key = <Client-to-TGS Session Key>	
	last-req	
	nonce	
	Optional - key-expiration	
	flags	
	authtime	
ASRepPart encrypted with <User Secret Key> or <AS-Rep Encryption Key> if PKINIT	Optional - starttime	
	endtime	
	Optional - renew-till	
	srealm	
	sname	
	Optional - caddr	
	Optional - encrypted_pa_data	

Client

Parse KRB_AS_REP

PKINIT Authentication	Decode PA-PK-AS-REP (17) Value with DecodeData (<Kerberos->as-rep->padata->PA-DATA PA-PK-AS-REP-> padata-type->padata-value>, <DHPrivKey>, <DHParams>, < DHNonce>) and obtain <AS-Rep Encryption Key>
Decrypt ASRepPart	ASRepPart = Decryption (<EncType>, <User Secret Key> or <AS-Rep Encryption Key> if PKINIT, 3, <Kerberos->as-rep->enc-part->cipher>) and obtain <Client-to-TGS Session Key>

Key Distribution Center (KDC)

Parse KRB_TGS_REQ

Decrypt TGT TicketPart with Decryption (<EncType>, < Krbtgt Secret Key>, 2, Kerberos->tgs-req->padata->PA-DATA PA-TGS-REQ->padata-type->padata-value->ap-req->ticket->enc-part->cipher) and obtain <Client-to-TGS Session Key>

Decrypt Authenticator with <Client-to-TGS Session Key>

If <cname> in <Authenticator> != <cname> in <TGT>

Send KRB_TGS_REP

KRB_TGS_REQ

PADATA		Ticket Granting Ticket = TGT	
	Basic Request	PA-TGS-REQ (1)	Authenticator encrypted with Encryption (<EncType>, < Client-to-TGS Session Key>, 7, <Confounder>, EncodeData (<Realm>, <Cname>, <DateStr_UTC>, ...))
			authenticator-vno
			crealm
			cname
			Optional – cksum
			cusec
			ctime
			Optional – subkey
			Optional – seq-number
			Optional – authorization-data
		Ticket Granting Ticket = TGT	
	S4U2Self	PA-TGS-REQ (1) + PA-FOR-USER (129)	Authenticator encrypted with Encryption (<EncType>, < Client-to-TGS Session Key>, 7, <Confounder>, EncodeData (<Realm>, <Cname>, <DateStr_UTC>))
			userName = cname to impersonate
			userRealm
		PA-FOR-USER (129) Value = EncodeData (<Client-to-TGS Session Key>, 17, <Realm>, <CnameToImpersonate>)	cksum
			authPackage
		Ticket Granting Ticket = TGT	
	S4U2Proxy	PA-TGS-REQ (1) + PA-PAC-OPTIONS (167)	Authenticator encrypted with Encryption (<EncType>, < Client-to-TGS Session Key>, 7, <Confounder>, EncodeData (<Realm>, <Cname>, <DateStr_UTC>))
			PA-PAC-OPTIONS (167) Value = EncodeData (PA_PAC_ OPTIONS["flags"][resource_based_constrained_delegation = 3] = 1)
	kdc-options	If S4U2Proxy: kdc-options[cname_in_addl_tkt = 14] = 1; kdc-options[canonicalize = 15] = 1; kdc-options[forwardable = 1] = 1; kdc-options[renewable = 8] = 1	
	realm		
	sname	If S4U2Self + U2U -> sname is set with cname	
REQ-BODY	till		
	nonce		
	etype		
	S4U2Proxy	additional-tickets = Service Ticket from KRB_TGS_REP after KRB_TGS_REQ with S4U2Self	
	S4U2Self + U2U	additional-tickets = Ticket Granting Ticket = TGT from KRB_ AS_REP	

KRB_TGS_REP

crealm			
cname = cname to impersonate if S4U			
Service Ticket = ST	tkt-vno = Ticket format version number		
	realm		
	sname = cname if S4U2Self + U2U		
	etype		
		Flags = Ticket flags	
		Session Key = <Client-to-Service Session Key>	
		crealm	
		cname = cname to impersonate if S4U	
		transited	
		authtime	
		Optional – starttime	
		endtime	
		Optional – renew-till	
		Optional – caddr	
			LogonTime
			LogoffTime
			KickOffTime
			PasswordLastSet
			PasswordCanChange
			EffectiveName
			FullName
			LogonScript
			ProfilePath
			HomeDirectory
			HomeDirectoryDrive
			LogonCount
			BadPasswordCount
			UserId
			PrimaryGroupId
			GroupCount
		Optional – PAC_LOGON_INFO	GroupIds
			UserFlags
			UserSessionKey
			LogonServer
			LogonDomainName
			LogonDomainId
			LMKey
			UserAccountControl
			SubAuthStatus
			LastSuccessfulILogon
	<TicketPart> encrypted with <Service Secret Key> or <Client-to-TGS Session Key> if S4U2Self + U2U		LastFailedILogon
			FailedILogonCount
			SidCount
			ExtraSids
			ResourceGroupDomainSid
		Optional – authorization-data	ResourceGroupCount
			ResourceGroupIds
			ClientId
		Optional – PAC_CLIENT_INFO_TYPE	NameLength
			Name
			UpnLength
			UpnOffset
			UpnName
			DnsDomainNameLength
			DnsDomainNameOffset
			DnsName
		Optional – PAC_UPN_DNS_INFO	Flags
			Optional – SamNameLength
			Optional – SamNameOffset
			Optional – SamName
			Optional – SidLength
			Optional – SidOffset
			Optional – Sid
		Optional – PAC_SERVER_CHECKSUM	SignatureType
			Signature
		Optional – PAC_PRIVSVR_CHECKSUM	SignatureType
			Signature
			Version
			EncryptionType
		PKINIT Authentication – PAC_CREDENTIALS_INFO	CredentialCount
		SerializedData encrypted with <AS-Rep Encryption Key>	Version
			Credentials: Flags
			Credentials: LmPassword
			Credentials: NtPassword
		Optional – PAC_DELEGATION_INFO	
		Optional – PAC_ATTRIBUTES_INFO	
		Optional – PAC_REQUESTOR_INFO	
	Session Key = <Client-to-Service Session Key>		
	last-req		
	nonce		
	Optional – key-expiration		
	flags		
	authtime		
TGSRepPart encrypted with <Client-to-TGS Session Key>	Optional – starttime		
	endtime		
	Optional – renew-till		
	srealm		
	sname = cname if S4U2Self + U2U		
	Optional – caddr		
	Optional – encrypted_pa_data		

Client

Parse KRB_TGS_REP

S4U2Self + U2U	Decrypt <TicketPart> of <ServiceTicket> with Decryption (< EncType>, <AS-Rep Encryption Key>, 2, <Kerberos->tgs-rep->ticket->enc-part>)	Decrypt PAC_CREDENTIALS_INFO with Decryption (<AS-Rep Encryption Key>, 16, PAC_CREDENTIALS_INFO['SerializedData']) and obtain <NtPassword> of <cname>
Decrypt TGSRepPart	TGSRepPart = Decryption (<EncType>, <Client-to-TGS Session Key>, 8, <Kerberos->tgs-rep->enc-part->cipher>) and obtain <Client-to-Service Session Key>	

KRB_AP_REQ

ap-options = Flags and options	
Service Ticket = ST from KRB_TGS_REP	
Authenticator encrypted with Encryption (<EncType>, < Client-to-Service Session Key>, 7, <Confounder>, EncodeData (<Realm>, <Cname>, <DateStr_UTC>, ...))	authenticator-vno
	crealm
	cname
	Optional – cksum
	cusec
	ctime
	Optional – subkey
	Optional – seq-number
	Optional – authorization-data

Parse KRB_AP_REQ

Decrypt ST TicketPart with Decryption (<EncType>, < Service Secret Key>, 2, <<Protocol>->Session Setup Request->Security Blob->GSS-API->Simple Protected Negotiation->negTokenInit->krb5_blob->Kerberos->ap-req->ticket->enc-part->cipher>) and obtain <Client-to-Service Session Key>

Decrypt Authenticator with <Client-to-Service Session Key>

If <cname> in <Authenticator> != <cname> in <ST>

Optional – Send KRB_AP_REP

Service Server (SS)

<Protocol> communication

KRB_AP_REP

Timestamp (ctime + cusec) of Authenticator from KRB_AP_ REQ encrypted with <Client-to-Service Session Key>

Client