CMSC 426
Principles of Computer Security

Linux and Windows Authentication

Last Class We Covered

- Authentication
- Password Hashing
- Password Cracking
  - Brute-force attacks
  - Dictionary attacks
  - Rainbow tables
- Salted passwords

Any Questions from Last Time?

Today’s Topics

- Linux authentication
- Windows authentication
  - Standalone system authentication
  - Domain authentication
- Kerberos protocol

Linux Authentication

The /etc/passwd File

- Text file that stores information about user accounts
  - Readable by any user
  - On legacy UNIX systems, stored passwords hashed by the UNIX **`crypt()`** algorithm
  - Now a bit of a misnomer on modern Linux operating system

- For each user, lists username, user ID (uid), group ID (gid), comments, home directory, preferred shell

```
rj:x:1000:1000:RJ,,,:/home/rj:/bin/bash
```

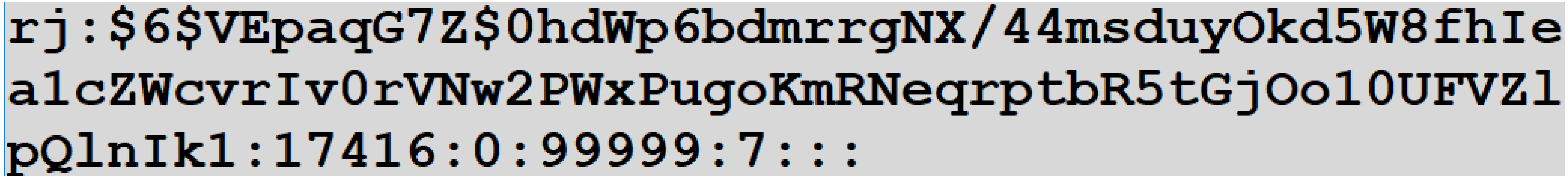
UNIX `crypt()` Algorithm

- UNIX passwords originally 8 characters max

- `crypt()` was a legacy function used to “hash” passwords
  - Password used was a 56-bit key
  - 12-bit salt
  - Uses a modified version of DES to make it one-way
  - Encrypts a 64-bit block of 0s, 25 rounds

The /etc/shadow File

- Text file that stores password information, only readable by root
- For each user, lists username, hashed password, info about password change/expiration policy
- Password field actually contains 3 separate parts separated by $
  - ID of hashing algorithm, salt, password hash



```
rj:$6$VEpaqG7Z$0hdWp6bdmrrgNX/44msduyOkd5W8fhIe
a1cZWcvrIv0rVNw2PWxPugoKmRNeqrptbR5tGjOo10UFVZl
pQlnIk1:17416:0:99999:7:::
```

Windows Authentication

Local Security Authority (LSA)

- Windows subsystem responsible for managing authentication and local security policy

- Local security policy determines:
  - Which users can access the system and in what way (*e.g.*, interactively, over the network, or as a service)
  - Which users have which permissions on the system
  - What forms of auditing are being performed

Security Accounts Manager (SAM)

- Database on standalone Windows systems that stores users’ password hashes

- Two password hashing algorithms have been used
  - Lan Manager hash (LanMan, LM)
  - NT hash (NTLM)

- On most modern Windows versions, the SAM file is encrypted to prevent offline password cracking
  - As of Windows 10, full disk encryption is preferred

Lan Manager Hash (LanMan, LM)

- In legacy versions of Windows, passwords were 14 characters max and not case sensitive

- LM hash algorithm:
  - Pad password to 14 characters
  - Convert to upper case
  - Split in half, use each half as a 56-bit DES key
  - DES encrypt the exact string "**`KGS!@#$%`**" with both keys
  - Concatenate the two encrypted strings

Lan Manager Hash Security

- Trivial to brute force
  - Just need to crack each of the two 7-byte halves
  - Exponentially easier to brute-force two 7-character strings than a single 14-character string
  - Also, since passwords are converted to uppercase, it's even easier to crack

- Default prior to Windows NT

- Disabled since Windows Vista

NT Hash (NTLM)

- Hash used by modern Windows systems

- NT Hash Algorithm:
  - Encode password in UTF-16 little-endian
  - Take the MD4 hash of the encoded password

- Other info:
  - Longer passwords now allowed
  - Passwords are still not salted

Windows Domains

- A group of computers connected over a network and managed by a central computer called a domain controller

- Password hashes stored in Active Directory database

- Three protocols have been used for authentication between a client and a server on a domain
  - NTLMv1 Protocol
  - NTLMv2 Protocol
  - Kerberos

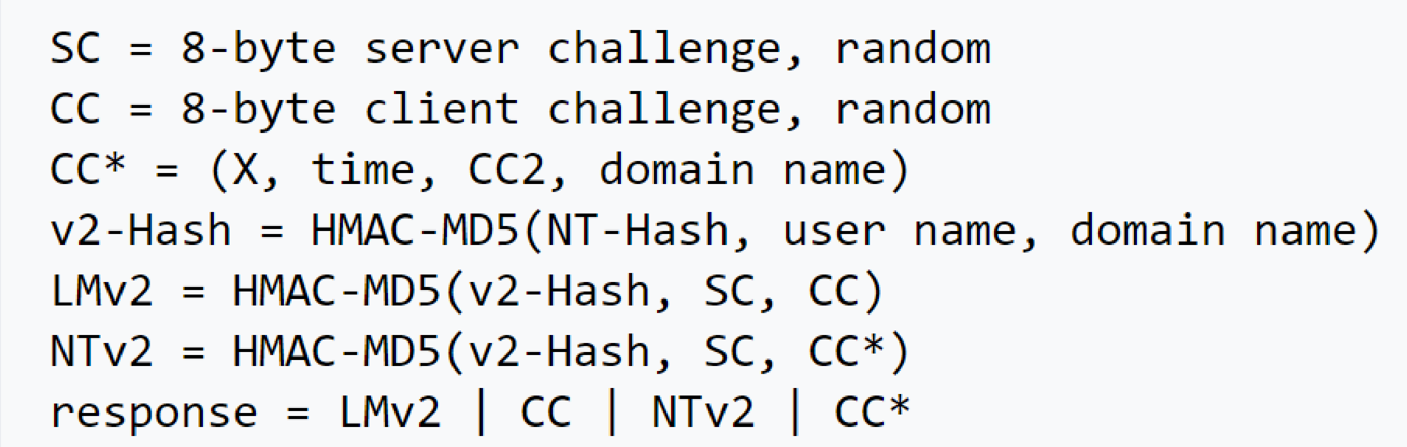
NTLMv1 Authentication Protocol

- Server issues a random 8-byte challenge to the client
- Client computes both the LM and NT hashes of the password
- Each 16-byte hash is padded to 21 bytes using 5 null bytes
- Both 21-byte values are separated into three 7-byte (56-bit) blocks
- Each block is DES encrypted using the challenge as a key, then all are appended together into a 48-byte response
- The server computes this as well and validates the response

- Insecure due to use of LM hash and DES

NTLMv2 Authentication Protocol

- Server issues a random 8-byte challenge to the client
- Client sends two responses containing information such as a random 8-byte value, the current time, the NT hash of the password, the user name, and the name of the domain
- Server validates responses



```
SC = 8-byte server challenge, random
CC = 8-byte client challenge, random
CC* = (X, time, CC2, domain name)
v2-Hash = HMAC-MD5(NT-Hash, user name, domain name)
LMv2 = HMAC-MD5(v2-Hash, SC, CC)
NTv2 = HMAC-MD5(v2-Hash, SC, CC*)
response = LMv2 | CC | NTv2 | CC*
```

Kerberos Protocol

Kerberos Protocol

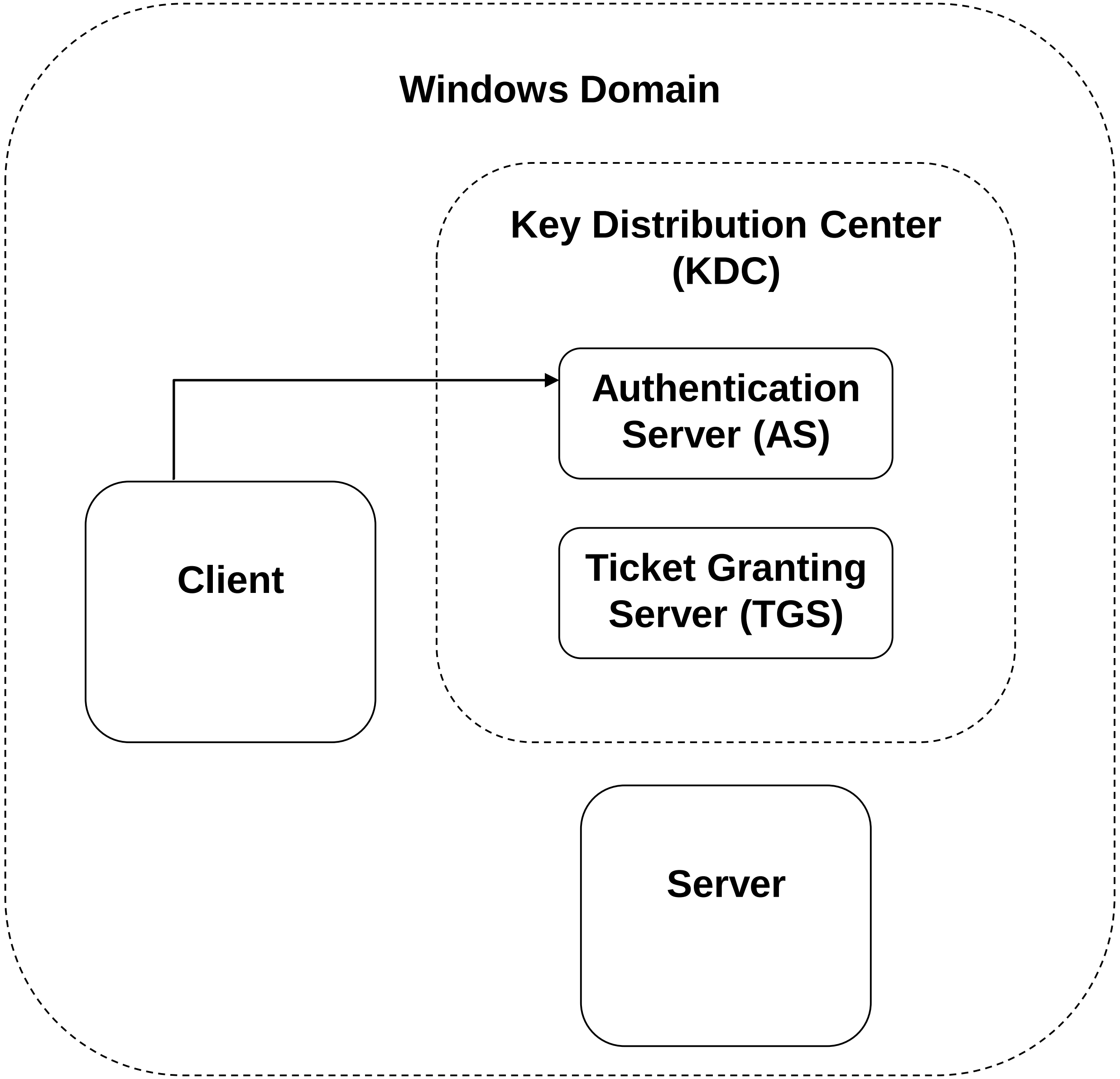
- Leading standard protocol for remote authentication
  - Used by many OSes, not just Windows
  - Will mostly be talking about it in the context of Windows domains

- Manages client-server interactions using a Key Distribution Center (KDC)

- Key Distribution Center provides two services:
  - Authentication Service (AS)
  - Ticket-Granting Service (TGS)

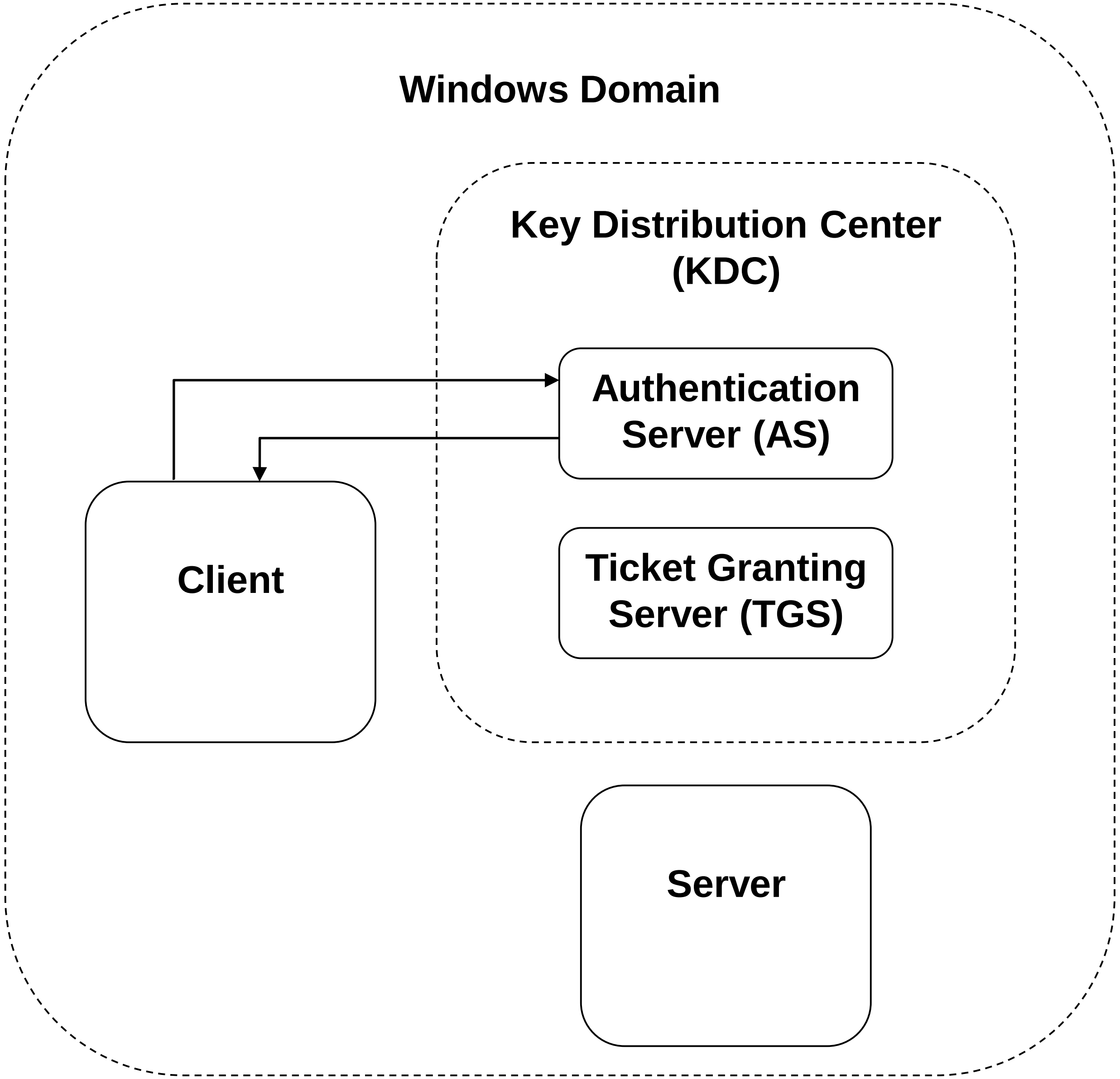
Kerberos Protocol

- Each time the client logs into a domain, they send their user ID and request for a Ticket-Granting Ticket (TGT) to the AS


Windows Domain
Key Distribution Center (KDC)
Authentication Server (AS)
Ticket Granting Server (TGS)
Client
Server

Kerberos Protocol

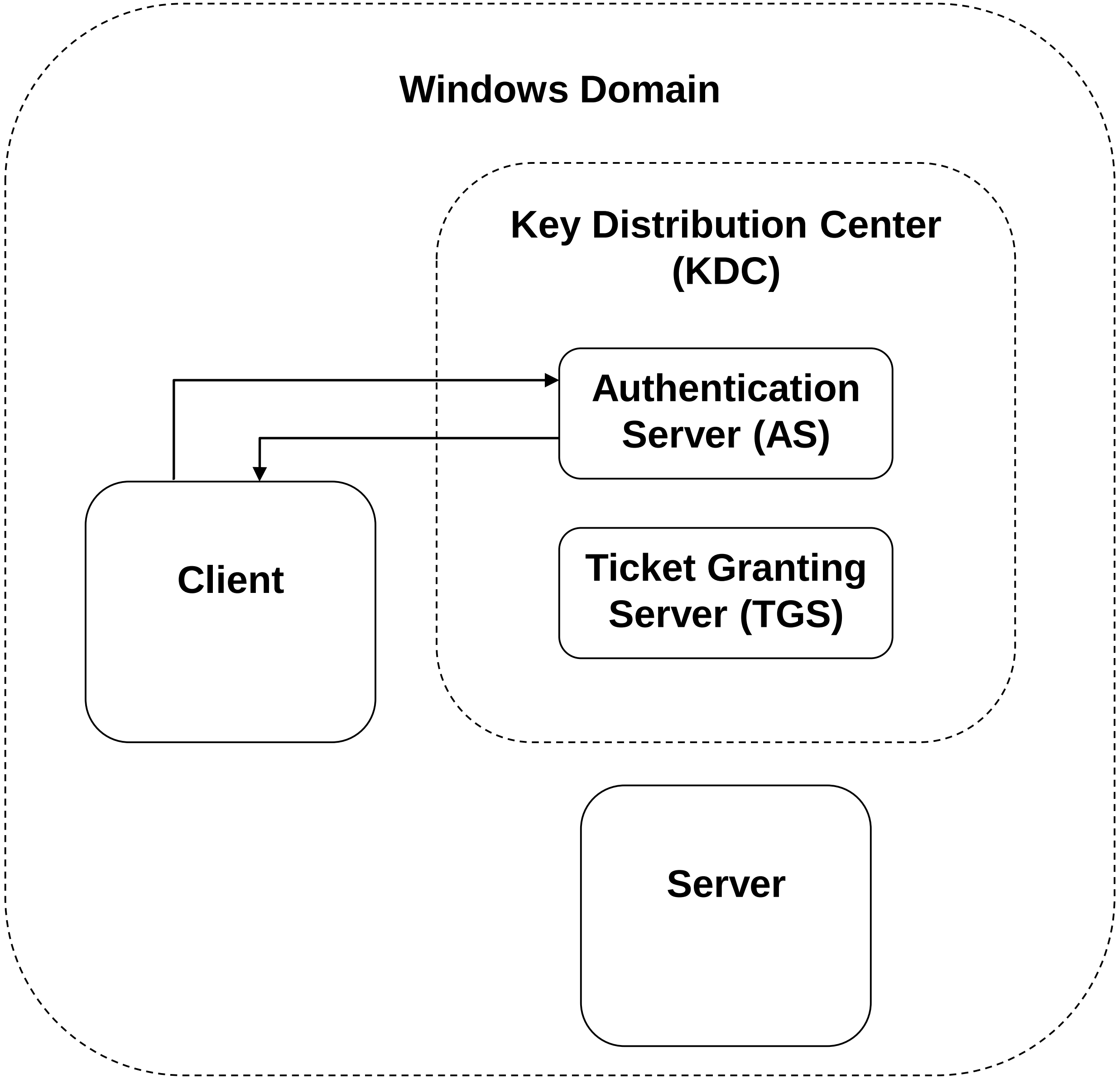
- The AS responds with two messages:
  - The TGT, which contains the user’s ID, TGS ID, timestamp, IP address, lifetime, and *TGS session key*, encrypted using the TGS secret key
  - The TGS ID, timestamp, lifetime, and *TGS session key*, encrypted using the client’s password hash as a key


Windows Domain
Key Distribution Center (KDC)
Authentication Server (AS)
Ticket Granting Server (TGS)
Client
Server

Kerberos Protocol

- The user enters their password and decrypts the second message

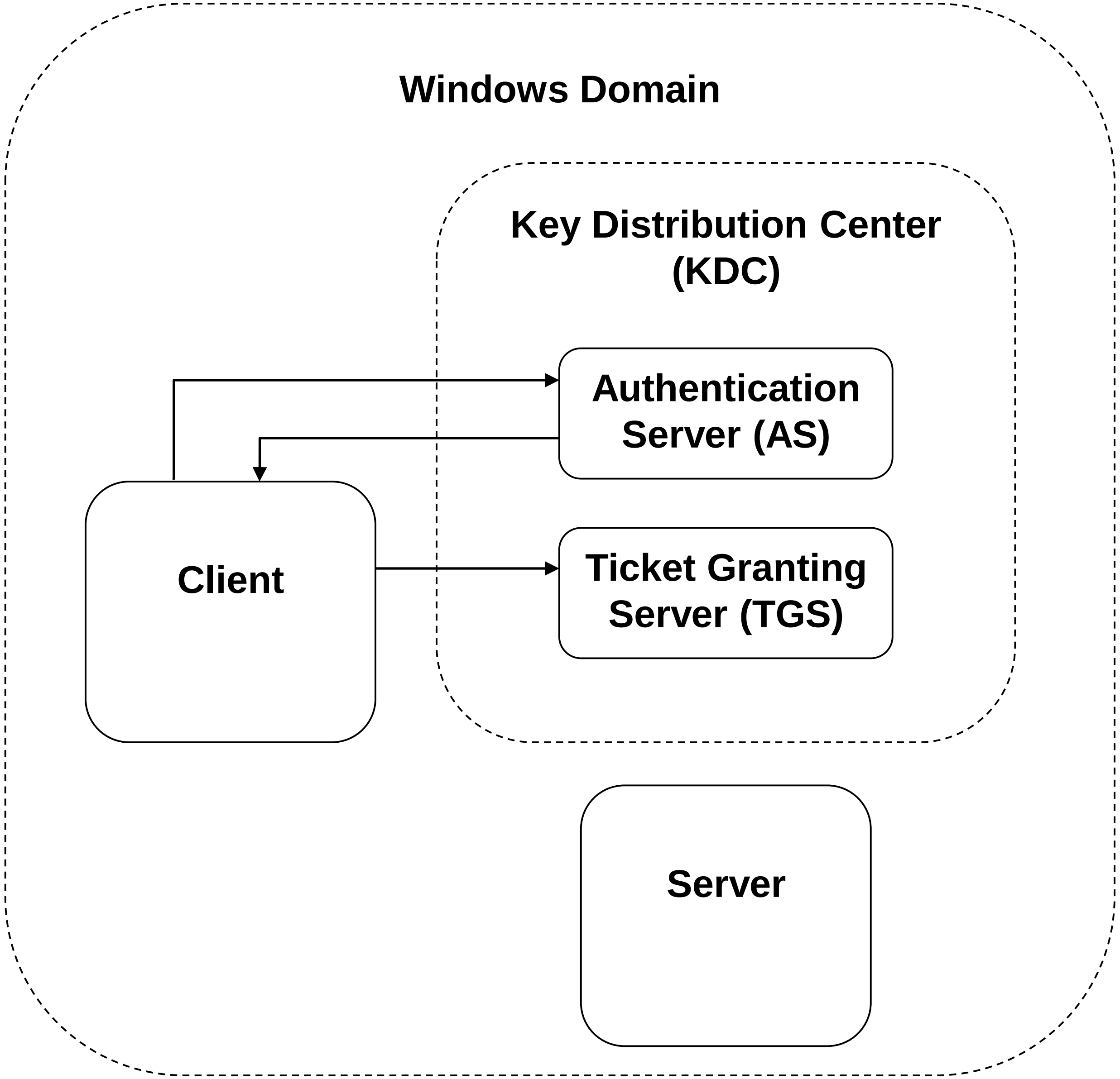
- The client prepares an *Authenticator*, which contains their user ID and timestamp, and encrypts it using the TGS session key


Windows Domain
Key Distribution Center (KDC)
Authentication Server (AS)
Ticket Granting Server (TGS)
Client
Server

Kerberos Protocol

- Any time the client needs to communicate with a server, it sends a message to the TGS requesting a ticket to the server

- The client also sends the encrypted TGT and encrypted Authenticator to the TGS

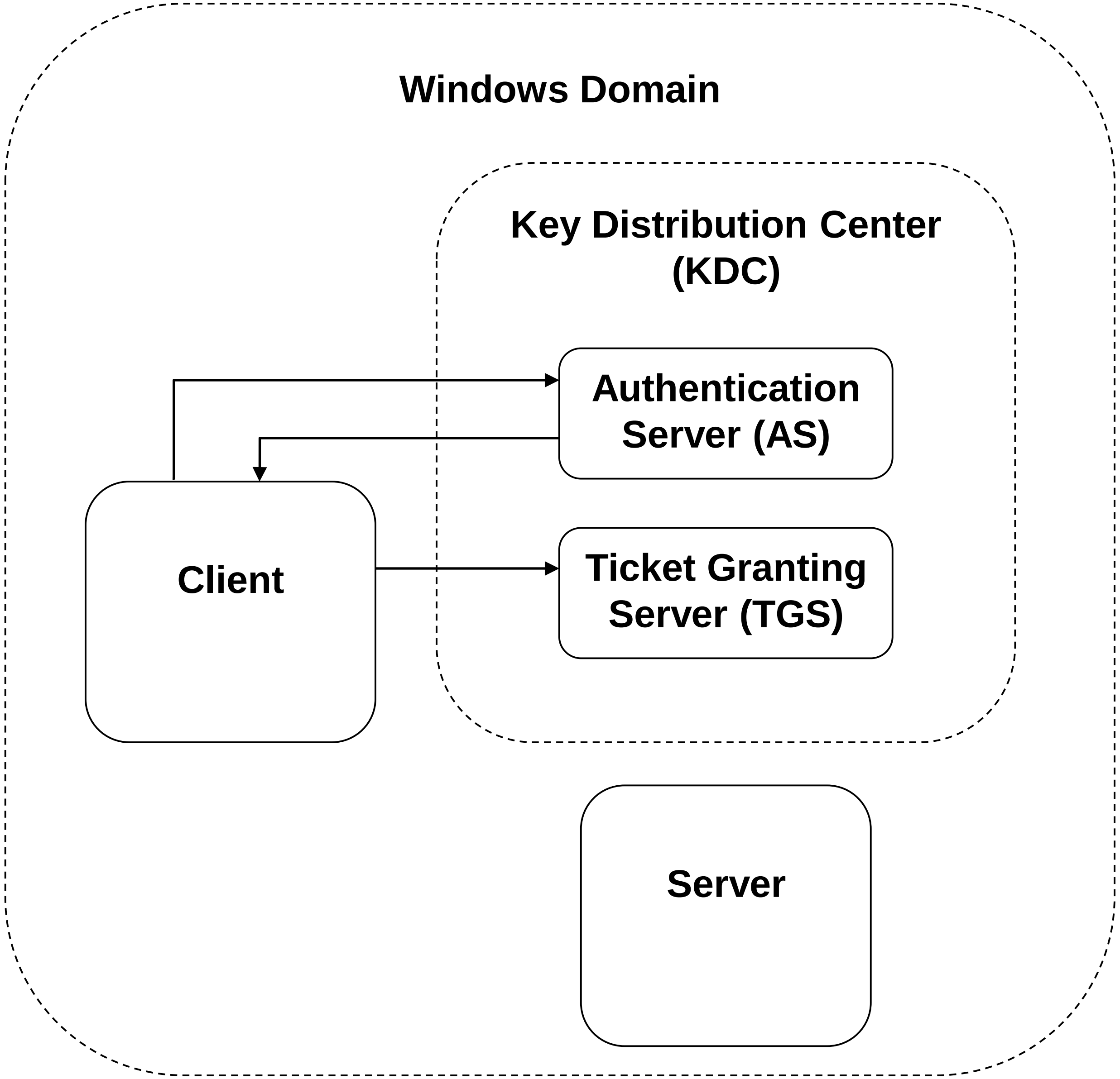

Windows Domain
Key Distribution Center (KDC)
Authentication Server (AS)
Ticket Granting Server (TGS)
Client
Server

Kerberos Protocol

- The TGS decrypts the TGT with its secret key

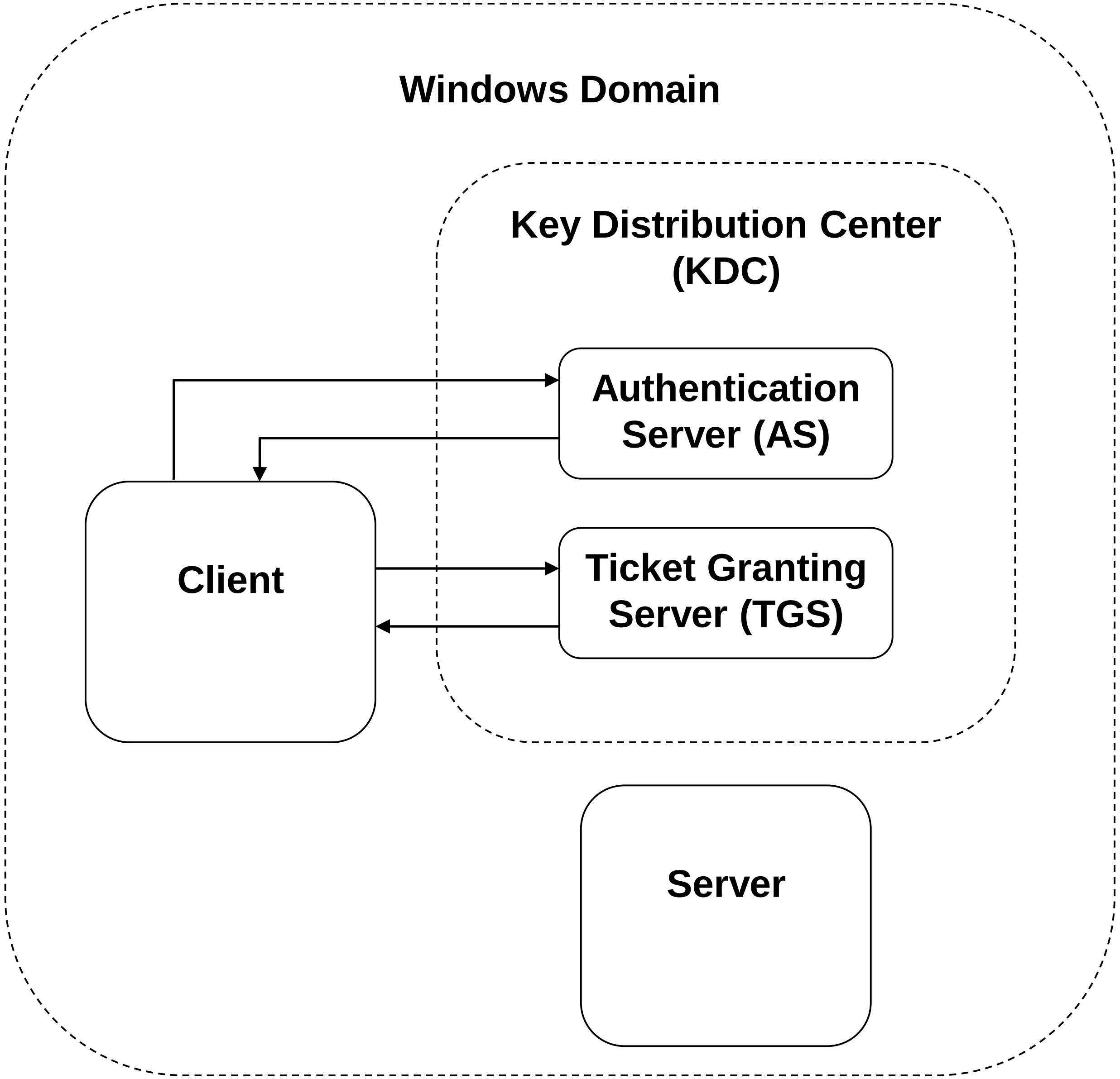
- The decrypted TGT contains the *TGS session key*, which the TGS uses to decrypt the Authenticator

- The TGS validates all information


Windows Domain
Key Distribution Center (KDC)
Authentication Server (AS)
Ticket Granting Server (TGS)
Client
Server

Kerberos Protocol

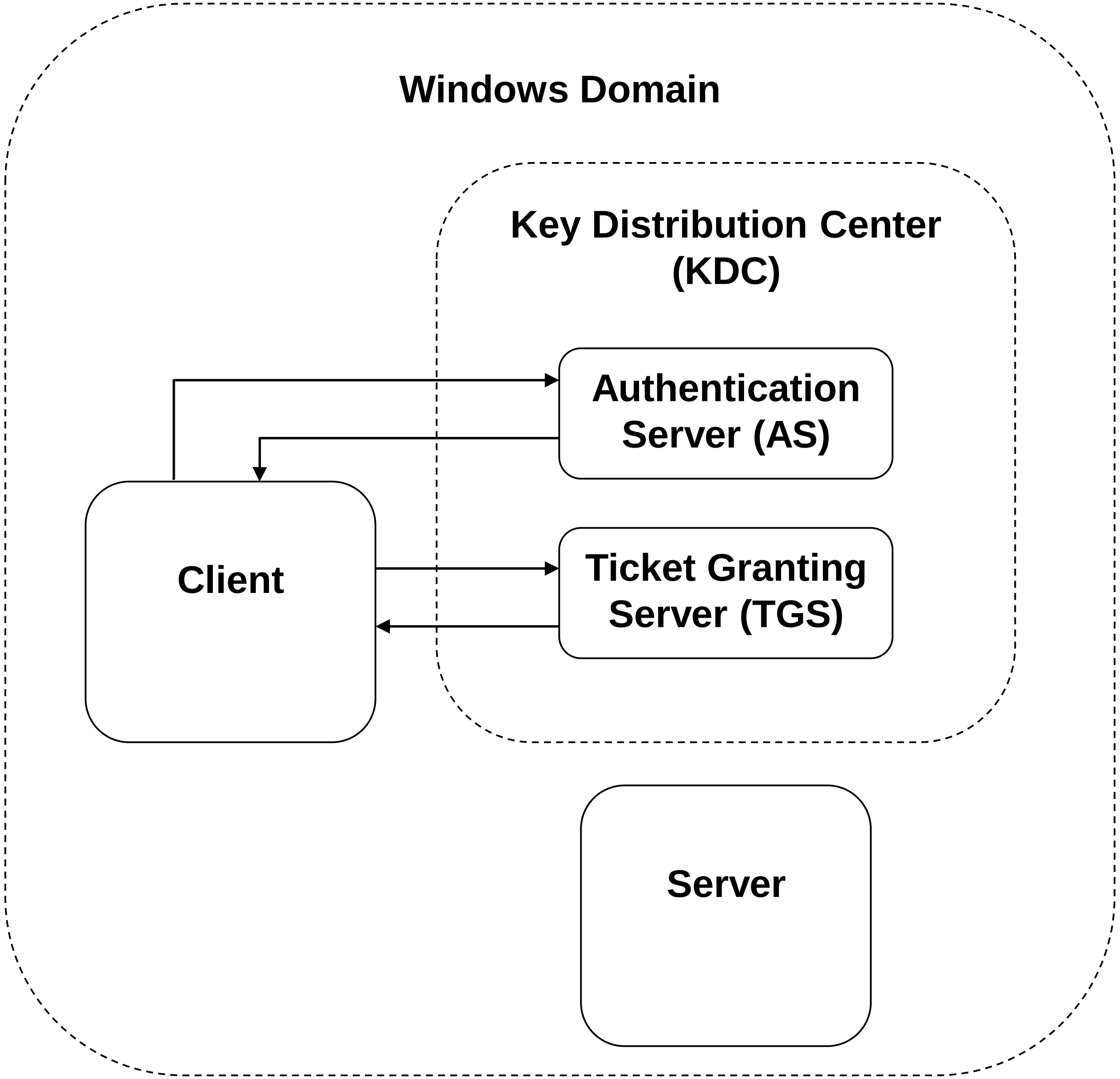
- The TGS sends two messages to the client:
  - A service ticket that contains the user’s ID, the service’s ID, IP address, timestamp, lifetime, and *service session key*, all encrypted using the service secret key
  - The service’s ID, timestamp, lifetime, and *service session key*, encrypted using the TGS session key


Windows Domain
Key Distribution Center (KDC)
Authentication Server (AS)
Ticket Granting Server (TGS)
Client
Server

Kerberos Protocol

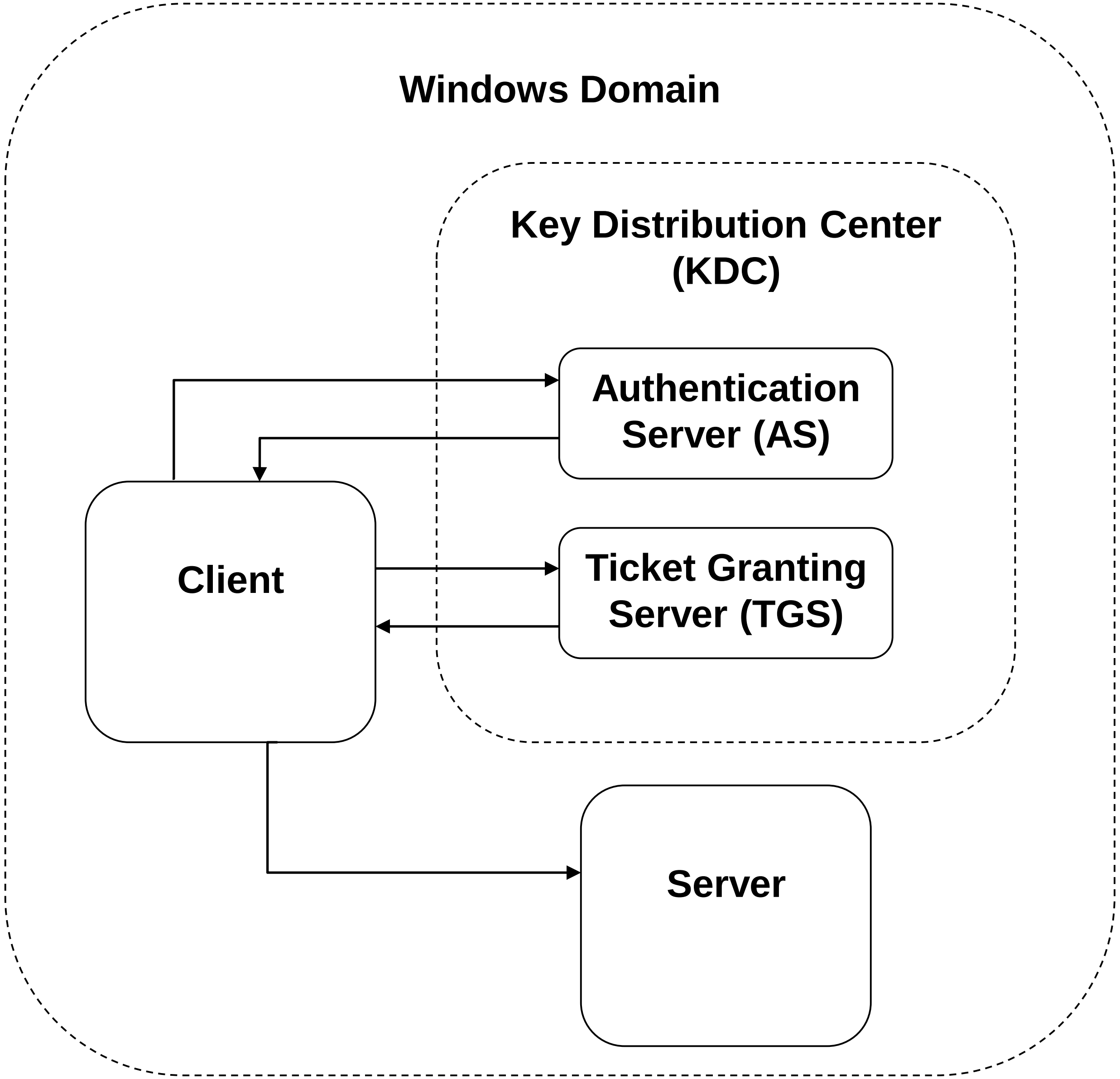
- The client decrypts the second message using the TGS session key

- The client prepares a ***second** Authenticator* that contains the user’s ID and timestamp and is encrypted using the service session key


Windows Domain
Key Distribution Center (KDC)
Authentication Server (AS)
Ticket Granting Server (TGS)
Client
Server

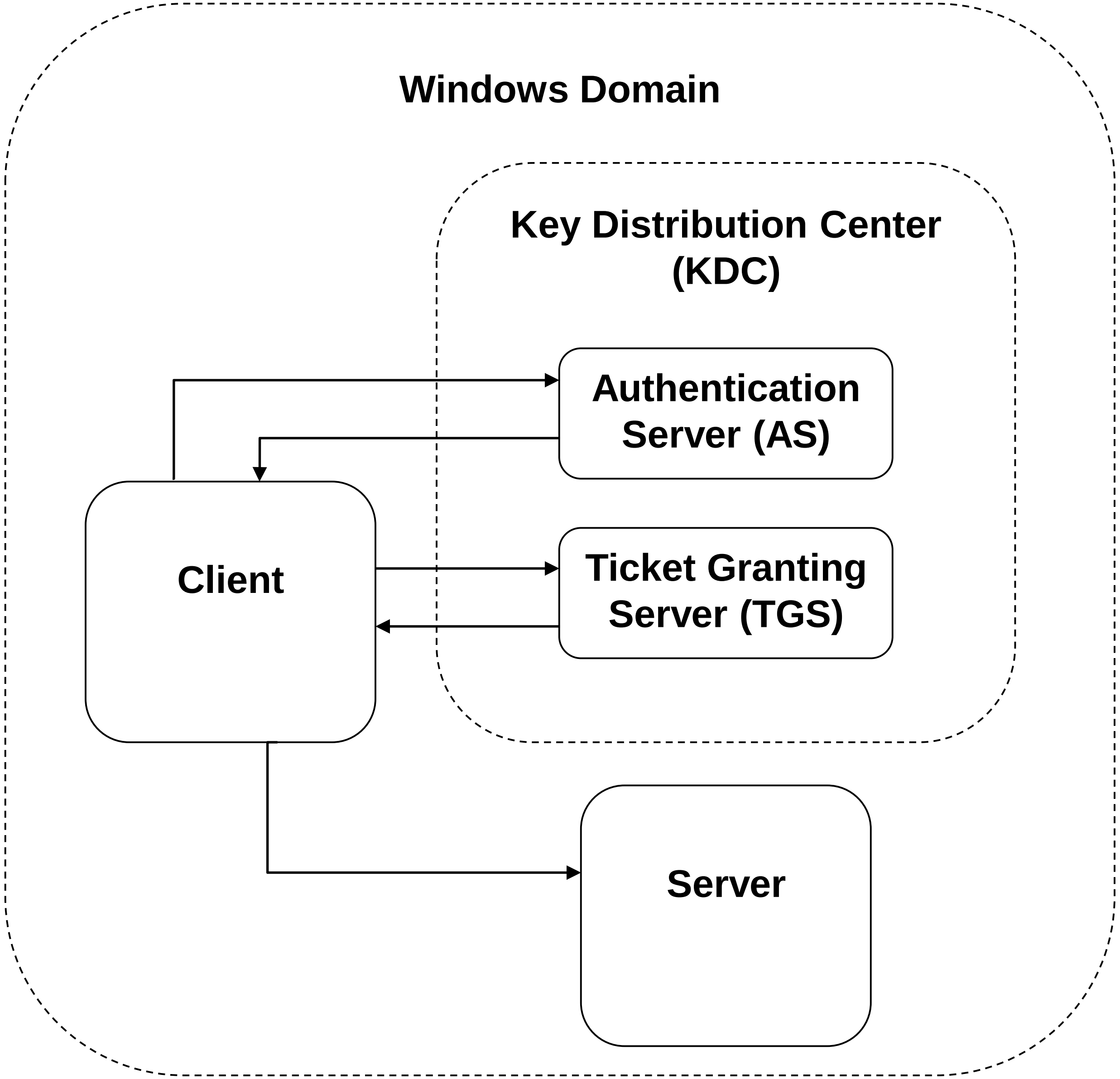
Kerberos Protocol

- The client sends the service ticket and the second Authenticator to the server


Windows Domain
Key Distribution Center (KDC)
Authentication Server (AS)
Ticket Granting Server (TGS)
Client
Server

Kerberos Protocol

- The server uses its secret key to decrypt the service ticket, which includes the *service session key*
- The server decrypts the second Authenticator with the service session key
- The server validates all information

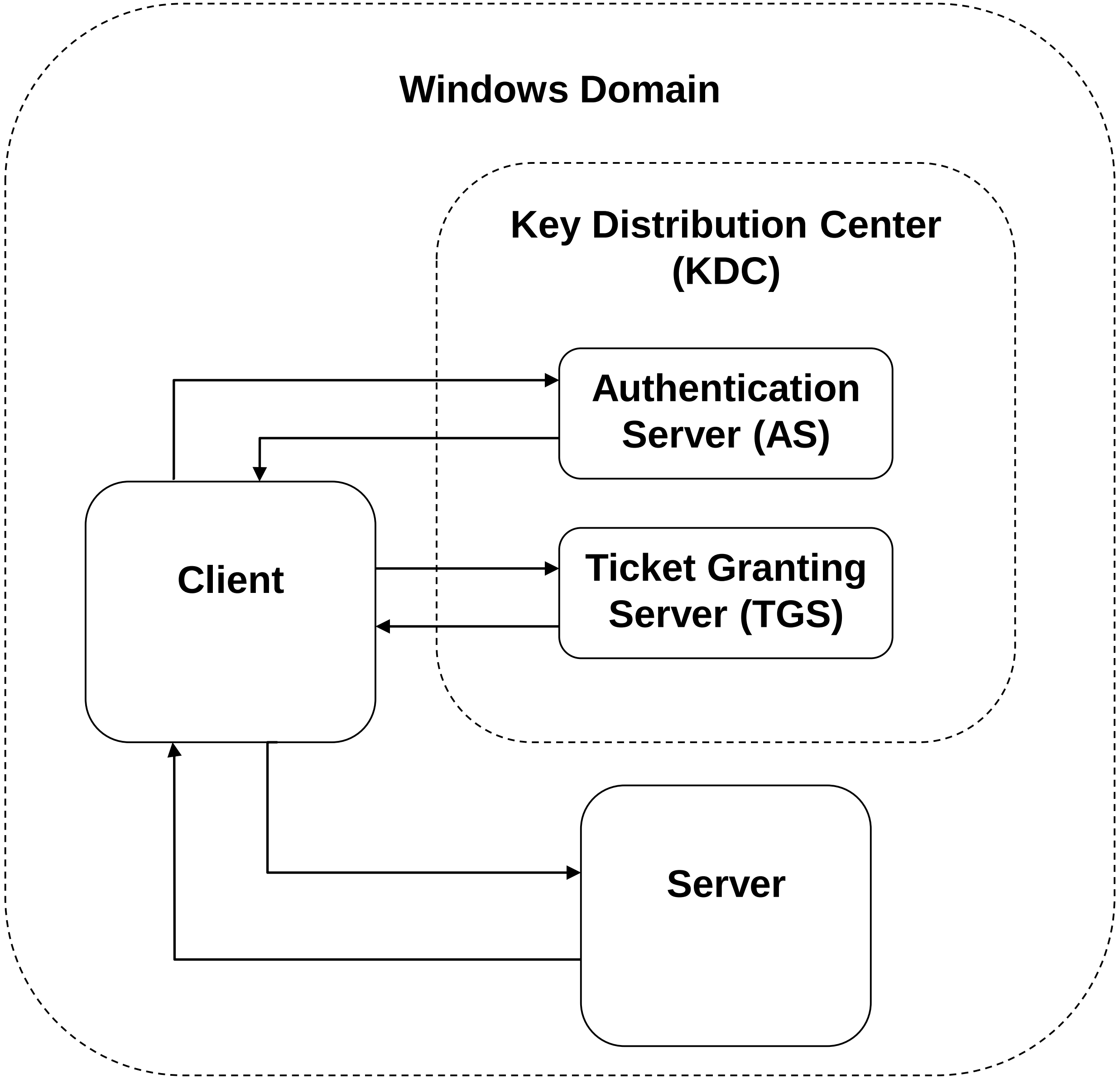

Windows Domain
Key Distribution Center (KDC)
Authentication Server (AS)
Ticket Granting Server (TGS)
Client
Server

Kerberos Protocol

- The server prepares a ***third** Authenticator* containing the server's ID and the timestamp, and encrypts it with the service session key

- The server sends its Authenticator to the client, which decrypts it

- Authentication complete!!!


Windows Domain
Key Distribution Center (KDC)
Authentication Server (AS)
Ticket Granting Server (TGS)
Client
Server

Announcements

- Next class will be Hardening
  - Topic will NOT be on the upcoming exam
  - Go vote!!!

- Midterm 2 is happening on Thursday (November 8th)