```
#!/bin/bash

# Check the number of arguments
if [ "$#" -ne 4 ]; then
  echo "Usage: $0 sender|receiver IP port source|destination"
  exit 1
fi

# Assign the arguments to variables
mode=$1
ip=$2
port=$3
path=$4

# Check the mode
if [ "$mode" = "sender" ]; then
  # Check if the source is a file or a directory
  if [ -f "$path" ]; then
    # Get the file name
    file=$(basename "$path")
    # Compress the file using gzip
    gzip -c "$path" > "$file.gz"
    # Encrypt and send the file using cryptcat
    cryptcat -k secret -w 2 "$ip" "$port" < "$file.gz"
    # Remove the compressed file
    rm "$file.gz"
    # Print a message
    echo "Sent $file to $ip:$port"
  elif [ -d "$path" ]; then
    # Get the directory name
    dir=$(basename "$path")
    # Compress the directory using tar and gzip
    tar czf "$dir.tar.gz" "$path"
    # Encrypt and send the file using cryptcat
    cryptcat -k secret -w 2 "$ip" "$port" < "$dir.tar.gz"
    # Remove the compressed file
    rm "$dir.tar.gz"
    # Print a message
    echo "Sent $dir to $ip:$port"
  else
    # Print an error message
    echo "$path is not a valid file or directory"
    exit 2
  fi
elif [ "$mode" = "receiver" ]; then
  # Check if the destination is a directory
  if [ -d "$path" ]; then
    # Receive and decrypt the file using cryptcat
    cryptcat -k secret -l -p "$port" > "$path/backup.gz"
    # Decompress the file using gzip
    gzip -d "$path/backup.gz"
    # Print a message
    echo "Received backup file from $ip:$port"
  else
    # Print an error message
    echo "$path is not a valid directory"
    exit 3
  fi
else
  # Print an error message
  echo "$mode is not a valid mode"
  exit 4
fi
```

Description: This script is called secure_remote_backup.sh and it uses Cryptcat to encrypt and transfer files or directories over the network. The script can run in two modes: sender or receiver. The sender mode compresses the source file or directory using gzip, encrypts it using a secret key, and sends it to the receiver's IP and port. The receiver mode listens on a port, decrypts the incoming file using the same secret key, and decompresses it to the destination directory. The script checks the number and validity of the arguments, and prints usage, error, or success messages accordingly.

To use the script, the sender and receiver machines must have Cryptcat and gzip installed. The sender and receiver must also agree on a secret key to use for encryption and decryption. The script takes four arguments: mode, IP, port, and path. The mode can be either sender or receiver. The IP is the IP address of the receiver machine. The port is the port number to use for communication. The path is the source file or directory for the sender, or the destination directory for the receiver.

For example,

Note: make sure to give the script file appropriate permission using the command: chmod 760 ./secure_remote_backup.sh.

You make a directory called backup in the machine where you want to save the receiving file.

Then you run the following command to start the backup server:
sudo ./secure_remote_backup.sh receiver 127.0.0.1 8080 ./backup/

On the sender side you would create a file called test.txt with the content of "Hello, world!". You can use the following command to backup the file to the receiver machine:
sudo ./secure_remote_backup.sh sender 127.0.0.1 8080 ./test.txt

This would backup the file to the backup server.