

```
// DSURF Pseudocode
```

```
procedure DSURF_Store(file)
```

```
    blocks ← BreakIntoBlocks(file) // Step 1: Split file into blocks
```

```
    pointers ← empty list // Initialize empty pointer list
```

```
    for each block in blocks do // Loop through each block
```

```
        random_location ← SelectRandomStorageLocation() // Step 2: Select random
storage location
```

```
        StoreBlockAt(random_location, block) // Step 3: Store block at random
location
```

```
        pointers.append(random_location) // Step 4: Add the pointer for later
retrieval
```

```
    end for
```

```
    StorePointersSecurely(pointers) // Step 5: Store pointers securely
```

```
end procedure
```

```
procedure DSURF_Retrieve(pointers)
```

```
    blocks ← empty list // Initialize list to hold retrieved blocks
```

```
    for each pointer in pointers do // Loop through each pointer
```

```
        block ← RetrieveBlockFrom(pointer) // Step 6: Retrieve block from
pointer location
```

```
        blocks.append(block) // Add retrieved block to the list
```

```
    end for
```

```
    file ← ReassembleBlocks(blocks) // Step 7: Reassemble blocks into the
original file
```

```
    return file // Return the reassembled file
```

```
end procedure
```

```
procedure SelectRandomStorageLocation()
```

```
    // Step 8: Select a random storage location using cryptographic randomness
```

```
    location ← GenerateRandomString() // Generate a random string for the
location
    return location
end procedure
```

```
procedure StoreBlockAt(location, block)
    // Step 9: Store the block at the specified location
    cloud_storage[location] ← block // Store block in cloud storage
end procedure
```

```
procedure StorePointersSecurely(pointers)
    // Step 10: Store pointer data securely in key management system
    secure_storage[pointers] ← encrypted // Encrypt and store pointers
end procedure
```

```
procedure RetrieveBlockFrom(pointer)
    // Step 11: Retrieve block from cloud storage using pointer
    block ← cloud_storage[pointer] // Retrieve the block at the given pointer
    return block
end procedure
```

```
procedure ReassembleBlocks(blocks)
    // Step 12: Reassemble blocks into the original file
    file ← ConcatenateBlocks(blocks) // Concatenate all blocks into a file
    return file // Return the reassembled file
end procedure
```