**1. Communication Process:** a. **Key Exchange:** - Use a key exchange protocol like Diffie-Hellman to securely establish a shared secret key between Alice and Bob. This ensures that an eavesdropper like Eve can't easily determine the key.

b. **Data Integrity (Condition 1):** - Apply a hash function (e.g., SHA-256) to the messages exchanged. Both Alice and Bob can hash their messages and compare the results. Any alteration by a third party, like Eve, would be detected.

c. **Authentication (Condition 2):** - Use digital signatures with public-key cryptography (e.g., RSA). Alice can sign her messages with her private key, and Bob can verify the signature using Alice's public key, ensuring the message is from her.

d. **Resource Constraints (Condition 3):** - Utilize lightweight cryptographic algorithms that are efficient in terms of memory and CPU usage. For example, consider using Elliptic Curve Cryptography (ECC) for key exchange.

e. **Secure File Sharing (Condition 4):** - Implement a symmetric encryption algorithm (e.g., AES) for encrypting the shared documents. The shared secret key from the key exchange can be used for symmetric encryption to ensure confidentiality.

f. **128-bit Security (Condition 5):** - Choose cryptographic algorithms with a sufficient key length, such as using a 256-bit key for AES, and elliptic curves with 256-bit key length for ECC.

**2. Cryptographic Tools:**

* **Key Exchange:** Diffie-Hellman
* **Data Integrity:** SHA-256
* **Authentication:** RSA for digital signatures
* **Resource Constraints:** Elliptic Curve Cryptography (ECC)
* **File Sharing:** AES for symmetric encryption

**3. Reasons for Cryptographic Tool Choices:**

* **Diffie-Hellman:** Efficient key exchange without needing a pre-shared key.
* **SHA-256:** Provides a strong and fast hash function for ensuring data integrity.
* **RSA:** Well-established for digital signatures and authentication.
* **ECC:** Efficient for key exchange in resource-constrained environments.
* **AES:** Fast and secure symmetric encryption, widely used.

**4. System Security:**

* The selected cryptographic tools provide at least 128-bit security due to their key lengths, ensuring resistance against brute-force attacks.

**5. Communication Scheme:**

Certainly! While I can't draw diagrams directly, I can provide you with a textual representation of the communication scheme involving Alice, Bob, Eve, and the cryptographic tools. You can use this description to create your own diagram.

**Communication Scheme:**

1. **Key Exchange (Diffie-Hellman):**
   * Alice generates her private key *a* and computes A = g^a mod  p.
   * Bob generates his private key *b* and computes *B*=*g^b* mod*p*.
   * They exchange *A* and *B*, allowing each to compute the shared secret *s*=*B^a* mod*p* = *A^b* mod*p*.
2. **Data Integrity (SHA-256):**
   * Before sending messages, Alice and Bob hash their messages using SHA-256.
   * Example: H(\text{"Hello from Alice"}) = \text{Hash\_Alice}.
3. **Authentication (RSA):**
   * Alice signs her messages with her private key, and Bob verifies using Alice's public key.
   * Example: \text{Signature\_Alice} = \text{Sign}(\text{"Hello from Alice"}, \text{Private Key\_Alice}).
4. **Resource Constraints (ECC):**
   * ECC is used for efficient key exchange, providing security with smaller key sizes compared to traditional methods.
5. **Secure File Sharing (AES):**
   * Alice encrypts the document with a symmetric key (AES), derived from the shared secret obtained through Diffie-Hellman.
   * Example: \text{Encrypted\_Document} = \text{AES\_Encrypt}(\text{Document}, \text{Shared\_Secret}).
6. **Communication Steps:** a. Alice initiates communication, performs Diffie-Hellman, and sends *A* to Bob. b. Bob receives *A*, performs Diffie-Hellman, and sends *B* to Alice. c. Both compute the shared secret *s*. d. Alice signs her messages with RSA and sends them to Bob. e. Bob verifies the signature and hashes the received message. f. Both exchange necessary information for file sharing. g. Alice encrypts the document and sends it to Bob. h. Bob decrypts the document using the shared secret.

**Diagram (Textual Representation):**

Diffie-Hellman SHA-256 RSA

--------------- --------- ---------

Alice a, A = g^a mod p Hash\_Alice Signature\_Alice

|----------------| |-----------------| |------------------|

\ /

\ /

\ /

v v

Bob b, B = g^b mod p Hash\_Bob Verification

|----------------| |------------------|

\ /

\ /

\ /

v v

Document Encrypted\_Document

**6. Security Against Attacks:** a. **Man-in-the-Middle Attack:** - Diffie-Hellman key exchange helps prevent this, as the shared key is not exchanged directly.

b. **Brute-Force Attack on AES:** - The 128-bit key length of AES ensures resistance against brute-force attacks.

**7. Attacks on the Scheme:**

* Illustrate potential man-in-the-middle and brute-force attacks on the diagram.

**8. System Name:**

* SecureComm 128

This is a high-level overview, and you may need to delve into specific details based on the cryptographic tools and protocols you choose. Additionally, make sure to consider implementation details and potential vulnerabilities in your system.