💡解答

为实现由 A 发往 B 的消息 M 的保密性、完整性和消息源完整，我们可以采取以下步骤：

首先，对于保密性，我们可以使用加密技术。比如，采用对称加密算法（如 AES）或非对称加密算法（如 RSA）对消息 M 进行加密。这一步的理由是，加密后的消息在传输过程中，即使被第三方获取，由于没有正确的密钥，第三方也无法理解消息的真实内容，从而保证了消息的保密性。

其次，关于完整性，我们可以使用消息认证码（MAC）或数字签名技术。比如，使用哈希函数（如 SHA-256）对消息 M 生成一个固定长度的摘要，然后使用私钥对摘要进行加密，形成数字签名。接收方 B 收到消息后，重新计算摘要并与接收到的数字签名进行比对。这一步的原因是，如果消息在传输过程中被篡改，接收方重新计算的摘要将与发送方生成的数字签名不匹配，从而能够检测出消息的完整性是否遭到破坏。

最后，为保证消息源完整，即确认消息确实是由 A 发出的，我们依赖前面提到的数字签名。因为只有拥有 A 的私钥才能生成有效的数字签名，接收方 B 可以通过使用 A 的公钥对数字签名进行解密和验证，来确认消息源的完整性。

综上所述，通过加密实现保密性，通过消息认证码或数字签名实现完整性和消息源完整，能够满足 A 向 B 发送消息 M 的要求。

+-----------------+

| A（发送端） |

+-----------------+

|

| 消息M

V

+-----------------+

| 生成摘要 | ----> Digest（M）

+-----------------+

|

|

V

+-----------------+

| 数字签名 | ----> Sign（Digest（M））

+-----------------+

|

|

V

+-----------------+

| 加密消息 | ----> Encrypted（M + Sign（Digest（M）））

+-----------------+

|

| 发送

V

+-----------------+

| B（接收端） |

+-----------------+

|

| 接收

V

+-----------------+

| 解密消息 | ----> M + Sign（Digest（M））

+-----------------+

|

|

V

+-----------------+

| 验证数字签名 | ----> 验证是否来自A

+-----------------+

|

|

V

+-----------------+

| 验证完整性 | ----> 对比摘要

+-----------------+