

Bob



Alice



# Cryptography with Google Tink

Prof Bill Buchanan OBE, The Cyber Academy

<http://asecuritysite.com>

Eve



**CYBER**  
**ACADEMY**

# OpenSSL to Tink

OpenSSL  
Cryptography and SSL/TLS Toolkit



LOGJAM

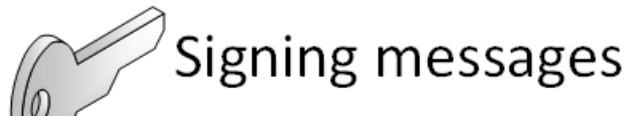
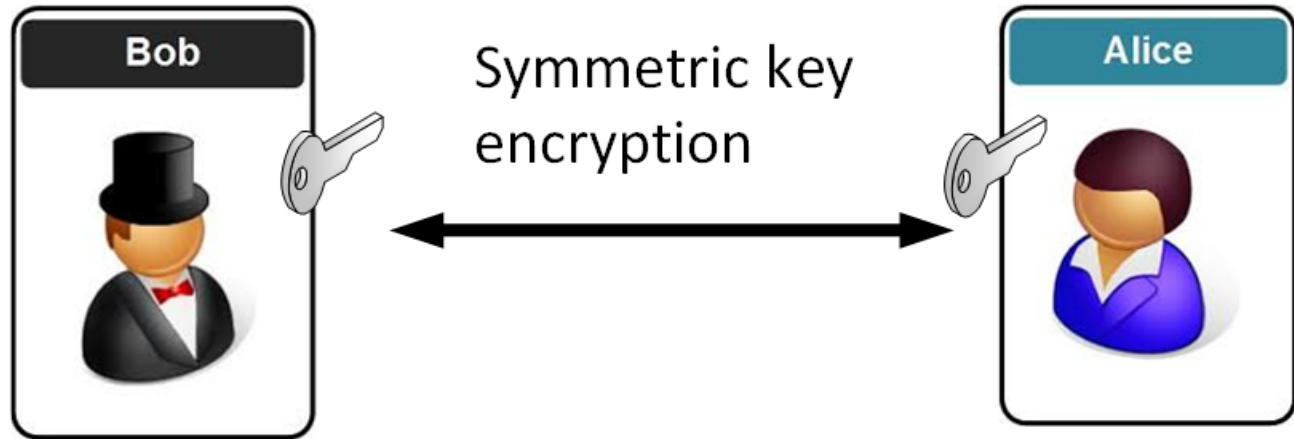


Forks:

LibreSSL  
BoringSSL

The screenshot shows the GitHub repository page for 'google/tink'. The browser address bar shows 'https://github.com/google/tink'. The repository name is 'google / tink'. It has 126 watchers, 2,242 stars, and 153 forks. The repository is licensed under Apache-2.0 and has 838 commits, 2 branches, 7 releases, and 25 contributors. The current branch is 'master'. A commit by 'thaidn and Tink Team' is shown, titled 'Fix golang copybara rule.' with the latest commit hash 'e699fe1' from 3 days ago. Another commit by 'apps' is shown, titled 'Update latest version references to 1.2.0.' from 24 days ago. A 'Join GitHub today' banner is visible at the top of the repository content area.

# Overview



Bob



Alice



# Cryptography with Google Tink: Symmetric Encryption

Prof Bill Buchanan OBE, The Cyber Academy

<http://asecuritysite.com>

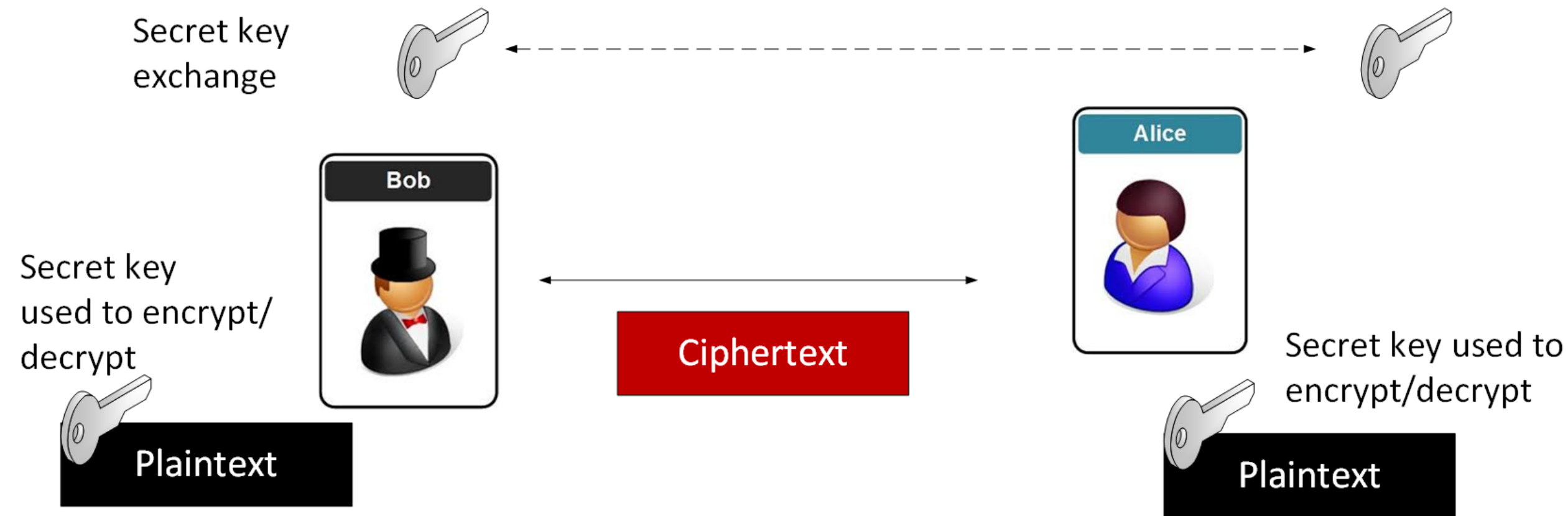
Eve



**CYBER**  
**ACADEMY**

# Symmetric Key with Tink

[Link](#)



[Link](#)

```
AeadConfig.register();  
try {KeysetHandle keysetHandle =  
KeysetHandle.generateNew(AeadKeyTemplates.AES128_GCM);  
Aead aead = AeadFactory.getPrimitive(keysetHandle);  
  
String plaintext="napier";  
String aad="qwerty123";  
System.out.println("Text:"+plaintext);  
byte[] ciphertext = aead.encrypt(plaintext.getBytes(), aad.getBytes());  
  
System.out.println("Cipher:"+ciphertext.toString());  
byte[] decrypted = aead.decrypt(ciphertext, aad.getBytes());  
String s = new String(decrypted);  
System.out.println("Text:"+s);.getBytes());
```



[Link](#)

```
AeadConfig.register();  
try {KeysetHandle keysetHandle =
```

```
KeysetHandle keysetHandle = KeysetHandle.newBuilder().setPlainText("Text: hello123")
```

```
    .setAead(AeadConfig.getInstance("AES128GCM"))  
    .setPassword("Password: qwerty")
```

```
    .setType(1)
```

```
    .build();  
String plaintext = "Enc type: 128-bit AES GCM Cipher: AQbLoE0ino8ofgrvuSSLOKTaYjdPc/
```

```
String aad = "ovwWznuMeYfjP+TO1fc6cn7DE=Cipher:"
```

```
System.out.println("Cipher: " + ciphertext.toString());  
byte[] ciphertext = "4151624C6F4530696E6F386F666772767553534C4F4B5461596A6450632F6F
```

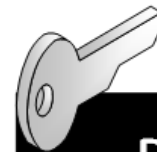
```
7677577A6E754D6559666A502B544F31666336636E3744453DDecrypted:  
hello123
```

```
System.out.println("Cipher:" + ciphertext.toString());
```

```
byte[] decrypted = aead.decrypt(ciphertext, aad.getBytes());
```

```
String s = new String(decrypted);
```

```
System.out.println("Text:" + s);  
}
```



Secret key used to  
encrypt/decrypt

**Plaintext**

[Link](#)

```
AeadConfig.register();  
try {KeysetHandle keysetHandle =  
KeysetHandle.newBuilder()  
Aead aead = AeadConfig.getInstance("AES-GCM").getAead();
```

```
String {  
String "primaryKeyId": 1525489658,  
System "key": [{  
byte "keyData": {  
System "typeUrl": "type.googleapis.com/google.crypto.tink.AesGcmKey",  
byte "keyMaterialType": "SYMMETRIC",  
String "value": "GhAmZA23Todsp2JPRj7u8130"  
System },  
"outputPrefixType": "TINK",  
"keyId": 1525489658,  
"status": "ENABLED"  
}]  
}
```



Bob



# Cryptography with Google Tink: Signing a Message

Alice



Prof Bill Buchanan OBE, The Cyber Academy

<http://asecuritysite.com>

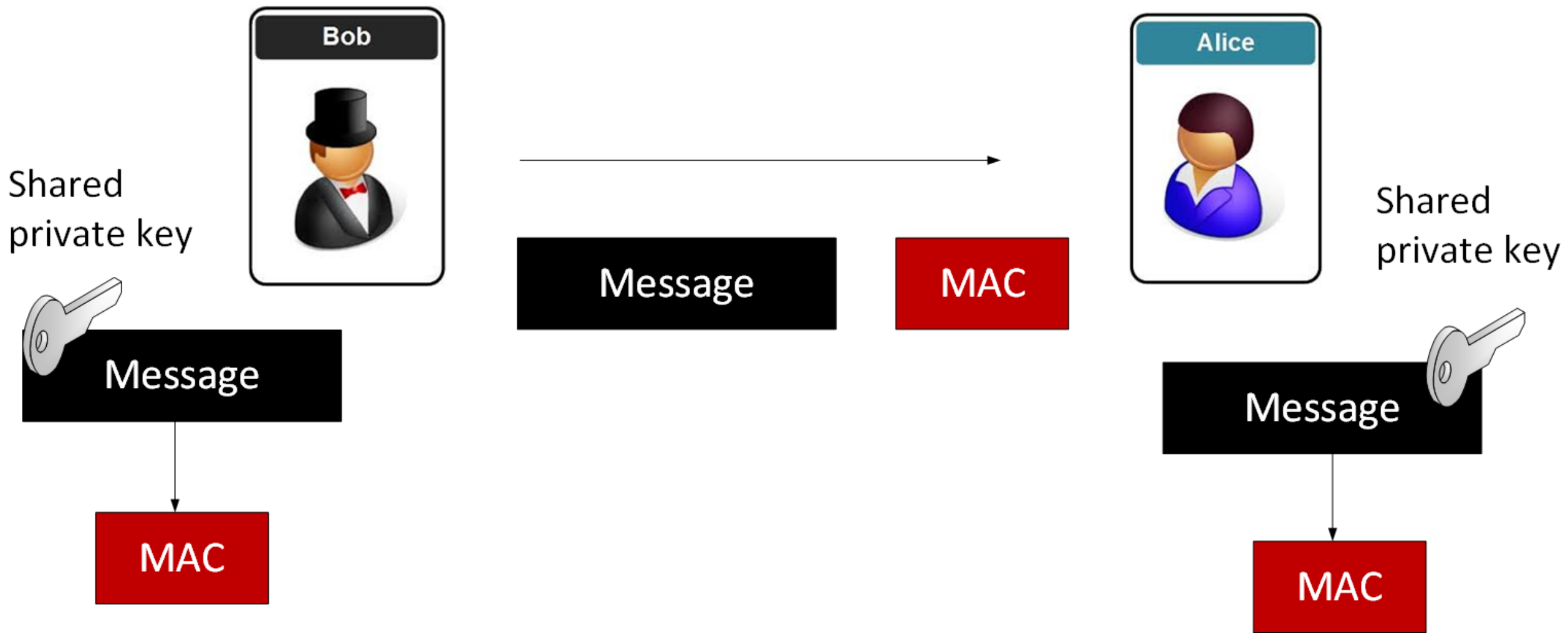
Eve



**CYBER  
ACADEMY**

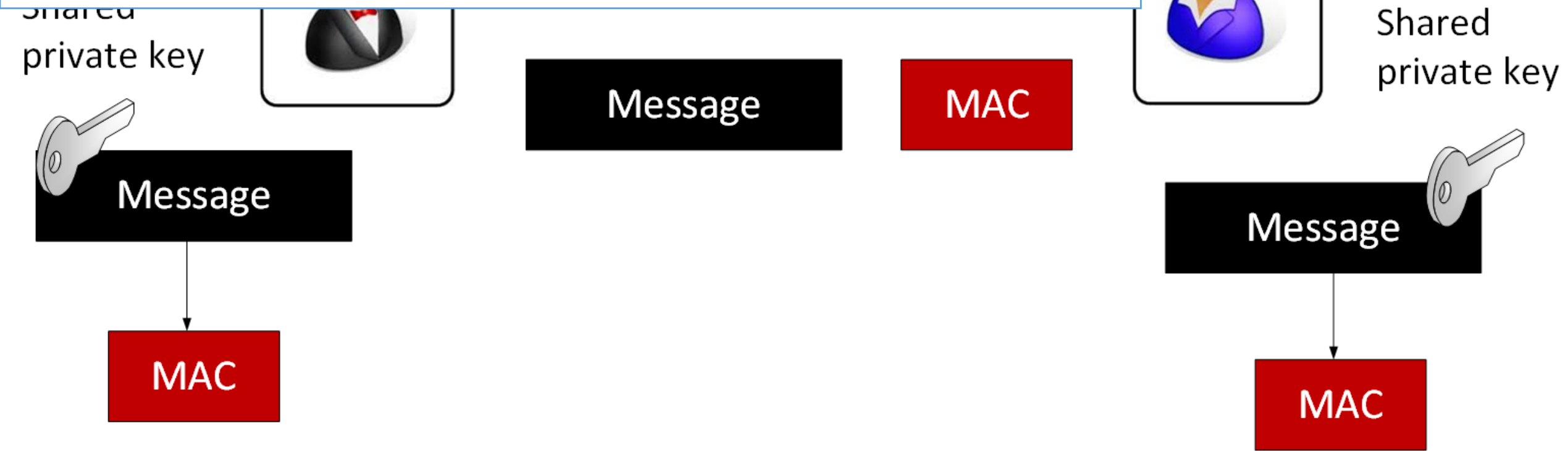
# Signing a Message with Tink

[Link](#)



[Link](#)

```
KeysetHandle keysetHandle =  
KeysetHandle.generateNew(MacKeyTemplates.HMAC_SHA256_128BI  
TTAG);  
Mac mac = MacFactory.getPrimitive(keysetHandle);  
byte[] tag = mac.computeMac(plaintext.getBytes());  
mac.verifyMac(tag,plaintext.getBytes());
```



[Link](#)

```
KeysetHandle keysetHandle =  
KeysetHandle.generateNew(MacKeyTemplates.HMAC_SHA256_128BI  
TTAG);
```

Alice

```
Mac mac = MacFactory.getPrimitive(keysetHandle).  
byte[] tag
```

Text: hello123

```
mac.verify
```

MAC: ASJFEIAQEqk9MvGalsJyKcLiN2iw

MAC: 41534A464549415145716B394D76476149734A794B634C694E326977

Valid MAC

Message

MAC

Message

MAC



[Link](#)

```
KeysetHandle keysetHandle =  
KeysetHandle.generateNew(MacKeyTemplates.HMAC_SHA256_128BI  
TTAG);
```

```
Mac mac = MacFactory.getPrimitive(keysetHandle).
```

```
byte[] tag = mac.computeMac("Text: hello123".getBytes());
```

```
MAC: {  
  "primaryKeyId": 574951552,  
  "key": [{  
    "keyData": {  
      "typeUrl": "type.googleapis.com/google.crypto.tink.HmacKey",  
      "keyMaterialType": "SYMMETRIC",  
      "value": "EgQIAxAQGIAU/p0/1SEV+01WE/fvQufi7z+rxQ0W6cJeRtgHHtqMQg=="  
    },  
    "outputPrefixType": "TINK",  
    "keyId": 574951552,  
    "status": "ENABLED"  
  }]  
}
```

Alice

Bob



# Cryptography with Google Tink: Digital Signing

Alice



Prof Bill Buchanan OBE, The Cyber Academy

<http://asecuritysite.com>

Eve



**CYBER  
ACADEMY**

# Signing With Tink

[Link](#)



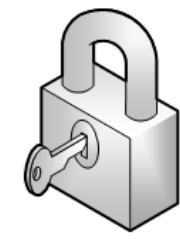
```
{
  "primaryKeyId": 835639698,
  "key": [{
    "keyData": {
      "typeUrl": "type.googleapis.com/google.crypto.tink.EcdsaPrivateKey",
      "keyMaterialType": "ASYMMETRIC_PRIVATE",
      "value": "EkwsBggDEAIYAhogafr30IV05SpchJZvJfQ6ChyWxhNIVSjpmztkJ+wYRusiIHCl4fYPSpuISdC18N90hWQ9jZvky6B0ytnL97HqafbrGiEAXJzrrM7H30NlzTRMdoPwWJ/PwkaQ2rCsUK5i3wGZMZQ="
    },
    "outputPrefixType": "TINK",
    "keyId": 835639698,
    "status": "ENABLED"
  }]
}
```

```
KeysetHandle privateKeysetHandle =  
KeysetHandle.generateNew(SignatureKeyTemplates.ECDSA_P256);
```

```
PublicKeySign signer =  
PublicKeySignFactory.getPrimitive(privateKeysetHandle);  
byte[] signature = signer.sign(plaintext.getBytes());
```

[Link](#)

Bob's Public  
key



Alice checks the

```
{  
  "primaryKeyId": 835639698,  
  "key": [{  
    "keyData": {  
      "typeUrl": "type.googleapis.com/google.crypto.tink.EcdsaPrivateKey",  
      "keyMaterialType": "ASYMMETRIC_PRIVATE",  
      "value": "EkwsBggDEAIYAhogafR30IV05SpchJZvJfQ6ChyWxhNIVSjpmztkJ+wYRusiIHCl4fYPSpuISdC18N90hWQ9jZvky6B  
0ytnL97HqafbrGiEAXJzrrM7H30NlzTRMdoPwWJ/PwkaQ2rCsUK5i3wGZMZQ="
```

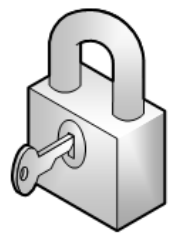


```
KeysetHandle privateKeysetHandle =  
KeysetHandle.generateNew(SignatureKeyTemplates.ECDSA_P256);
```

[Link](#)

```
PublicKeySign signer =  
PublicKeySignFactory.getPrimitive(privateKeysetHandle);
```

Bob's Public  
key



```
byte[] signature
```

```
Text: hello  
Sig (Base64):  
ATHO2ZIwRAIgd+JV6SOM08i01AFsrGR8JLenLDWtPKzoWUDRh4tBqC8
```

Checks the  
with Bob's

```
{  
  "primaryKeyId": 835639698,  
  "key": [{  
    "keyData": {  
      "typeUrl": "type.googleapis.com/google.crypto.tink.EcdsaPrivateKey",  
      "keyMaterialType": "ASYMMETRIC_PRIVATE",  
      "value": "EkwsBggDEAIYAhogafR30IV05SpchJZvJfQ6ChyWxhNIVSjpmztkJ+wYRusiIHCl4fYPSpuISdC18N90hWQ9jZvky6B  
0ytnL97HqafbrGiEAXJzrrM7H30NlzTRMdoPwWJ/PwkaQ2rCsUK5i3wGZMZQ="
```

Bob



Alice



# Cryptography with Google Tink

Prof Bill Buchanan OBE, The Cyber Academy

<http://asecuritysite.com>

Eve



**CYBER  
ACADEMY**