Authentication and Data Integrity

Cryptography

School of Engineering and Technology
CQUniversity Australia

Prepared by Steven Gordon on 23 Dec 2021,
auth.tex, r1951

Contents

Attacks on Information Transfer

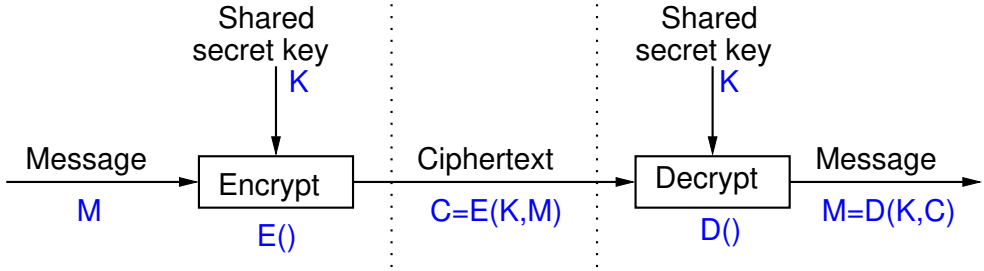
1. Disclosure: encryption
2. Traffic analysis: encryption
3. Masquerade: message authentication
4. Content modification: message authentication
5. Sequence modification: message authentication
6. Timing modification: message authentication
7. Source repudiation: digital signatures
8. Destination repudiation: digital signatures

Aims of Authentication

- Receiver wants to verify:
  1. Contents of the message have not been modified (*data authentication*)
  2. Source of message is who they claim to be (*source authentication*)
- Different approaches available:
  - Symmetric Key Encryption
  - Hash Functions
  - Message Authentication Codes (MACs)
  - Public Key Encryption (i.e. Digital Signatures)

Contents

Symmetric Encryption for Authentication



Recognising Correct Plaintext in English (question)

B receives ciphertext (supposedly from A, using shared secret key K):

`DPNFCTEJLYONCJAEZRCLASJTDQFY`

B decrypts with key K to obtain plaintext:

`SECURITYANDCRYPTOGRAPHYISFUN`

Was the plaintext encrypted with key K (and hence sent by A)? Is the ciphertext received the same as the ciphertext sent by A?

Recognising Correct Plaintext in English (question)

B receives ciphertext (supposedly from A, using shared secret key K):

`QEFPFPQEBTOLKDJBPPXDBPLOOVX`

B decrypts with key K to obtain plaintext:

`FTUEUEFTQIDAZSYQEEMSQEADDKM`

Was the plaintext encrypted with key K (and hence sent by A)? Is the ciphertext received the same as the ciphertext sent by A?

Recognising Correct Plaintext in Binary (question)

B receives ciphertext (supposedly from A, using shared secret key K):

01101001101011010101101110000010

B decrypts with key K to obtain plaintext:

01011101000011010010101001011110

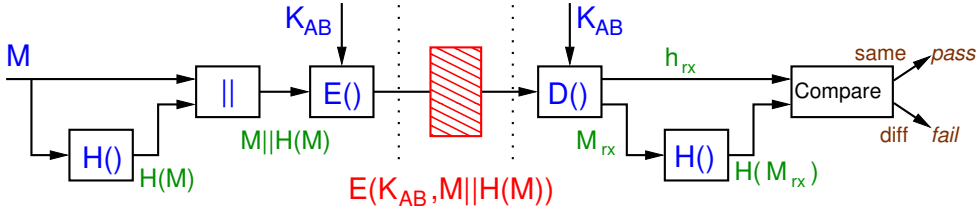
Was the plaintext encrypted with key K (and hence sent by A)? Is the ciphertext received the same as the ciphertext sent by A?

Recognising Correct Plaintext

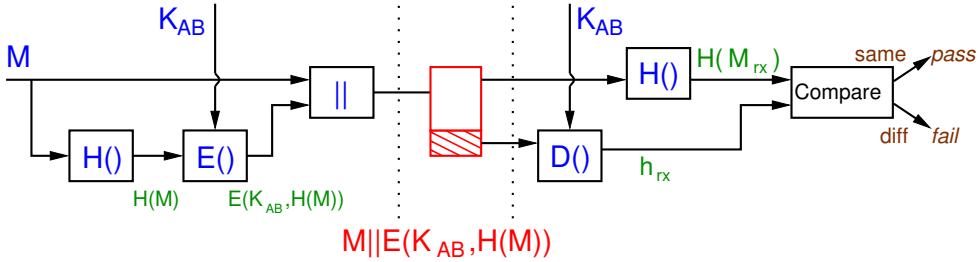
- Many forms of information as plaintext can be recognised at correct
- However not all, and often not automatically
- Authentication should be possible without decryptor having to know context of the information being transferred
- Authentication purely via symmetric key encryption is insufficient
- Solutions:
  - Add structure to information, such as error detecting code
  - Use other forms of authentication, e.g. MAC

Contents

Authentication by Hash and then Encrypt



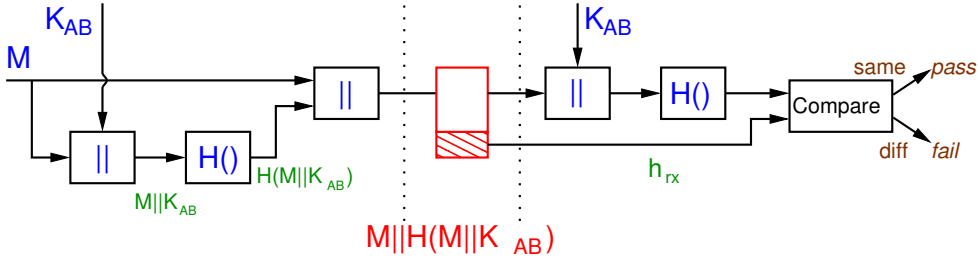
Authentication by Encrypting a Hash



Attack of Authentication by Encrypting a Hash (exercise)

If a hash function did not have the Second Preimage Resistant property, then demonstrate an attack on the scheme in The figure on slide 13.

Authentication with Hash of a Shared Secret

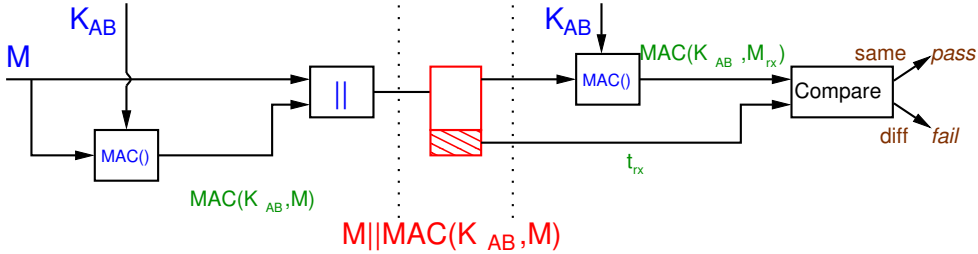


Attack of Authentication with Hash of Shared Secret (exercise)

If a hash function did not have the Preimage Resistant property, then demonstrate an attack on the scheme in The figure on slide 15.

Contents

Authentication with only MACs



Authentication using Encryption and a MAC

- Common to what both confidentiality and authentication (data integrity)
- MACs have advantage over hashes in that if encryption is defeated, then MAC still provides integrity
- But two keys must be managed: encryption key and MAC key
- Recommended algorithms used for encryption and MAC are independent
- Three general approaches (following definitions), referred to as authenticated encryption

Encrypt-then-MAC (definition)

The sender encrypts the message M with symmetric key encryption, then applies a MAC function on the ciphertext. The ciphertext and the tag are sent, as follows:

$$\mathrm{E}(K_1, M)||\mathrm{MAC}(K_2, \mathrm{E}(K_1, M))$$

Two independent keys, K_1 and K_2, are used.

MAC-then-Encrypt (definition)

The sender applies a MAC function on the plaintext, appends the result to the plaintext, and then encrypt both. The ciphertext is sent, as follows:

$$\mathrm{E}(K_1, M||\mathrm{MAC}(K_2, M))$$

Encrypt-and-MAC (definition)

The sender encrypts the plaintext, as well ass applying a MAC function on the plaintext, then combines the two results. The ciphertext joined with tag are sent, as follows:

$$\mathrm{E}(K_1, M)||\mathrm{MAC}(K_2, M)$$

Recommended Approach for Authenticated Encryption

- There are small but important trade-offs between encrypt-then-MAC, MAC-then-encrypt and encrypt-and-MAC
- Potential attacks on each, especially if a mistake in applying them
- Generally, encrypt-then-MAC is recommended, but are cases against it
- Some discussion of issues:
  - Chapter 9.6.5 of Handbook of Cryptography
  - Moxie Marlinspike
  - StackExchange
  - Section 1 and 2 of Authenticated Encryption by J Black
- Other authenticated encryption approaches incorporate authenticate into encryption algorithm
  - AES-GCM, AES-CCM, ChaCha20 and Poly1305

Contents

Digital Signatures

- Authentication has two aims:
  - Authenticate data: ensure data is not modified
  - Authenticate users: ensure data came from correct user
- Symmetric key crypto, MAC functions are used for authentication
  - But cannot prove which user created the data since two users have the same key
- Public key crypto for authentication
  - Can prove that data came from only 1 possible user, since only 1 user has the private key
- Digital signature
  - *Encrypt hash of message using private key of signer*

Digital Signatures in Practice

- User A has own key pair: (PU_A, PR_A)
- Signing
  - User A signs a message by encrypting hash of message with own private key: $S = E(PR_A, H(M))$
  - User attaches signature S to message M and sends to user B
- Verification
  - User B verifies a message by decrypting signature with signer's public key: $h = D(PU_A, S)$
  - User B then compares hash of received message, $H(M)$, with decrypted h; if identical, signature is verified

Digital Signature Example

User A	User B
Knows (PU_A,PR_A)	Knows PU_A
1. Sign message M: $S = E_{pub}(PR_A, H(M))$	
2. Append signature to message and send	
M || S	
	3. Decrypt $h = D_{pub}(PU_A, S)$
	4. Compare h with hash of received message
	if H(M) == h then message verified else verification failed (don't trust message)

In this example, the message is NOT confidential, but it is signed. If you require confidentiality AND signature, then must also encrypt the message (e.g. with symmetric key)