DUMPSBOSS.

WGU Introduction to Cryptography HNO1

WGU Introduction-to-Cryptography

Version Demo

Total Demo Questions: 11

Total Premium Questions: 111

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

QUESTION NO: 1

(Which technique involves spotting variations in encrypted data and plotting how the characters relate to standard English characters?)

A. Brute force

B. Frequency analysis

C. Known plaintext

D. Chosen ciphertext

ANSWER: B

Explanation:

Frequency analysis is the cryptanalytic technique that examines how often individual ciphertext characters, pairs of characters, or other patterns occur and compares those frequencies with the known statistical properties of a natural language such as English. English does not use every letter equally: letters such as E, T, and A, along with common sequences such as TH and HE, appear predictably often in typical text. In a cipher that consistently substitutes one symbol for another, those underlying patterns can remain visible in the encrypted output even though the actual letters have changed. An analyst can count and graph ciphertext-symbol occurrences, then use the resulting distribution to propose mappings between encrypted characters and likely English plaintext characters. This makes frequency analysis especially useful against classical monoalphabetic substitution ciphers and related schemes that do not adequately conceal language statistics. NIST describes cryptanalysis as the analysis of cryptographic systems to derive plaintext or secret information without authorized access, and frequency-based statistical analysis is a foundational historical example of that work. See NIST's Cryptanalysis Glossary and Britannica's overview of cryptanalysis.

QUESTION NO: 2

(Two organizations use an unauthenticated Diffie-Hellman exchange to establish a shared secret over the Internet. Which risk remains if no certificates, digital signatures, or pre-shared authentication values are used?)

A. A man-in-the-middle attack

B. A collision attack against the shared secret

C. A replay attack against the private key

D. A brute-force attack against the certificate chain

ANSWER: A

Explanation:

A man-in-the-middle attack is correct. Diffie-Hellman allows two parties to derive a shared secret over an untrusted network, but the basic exchange does not prove the identity of either participant. An attacker can establish one Diffie-Hellman secret with each organization, relay traffic between them, and decrypt or alter the communications.

Using sufficiently large parameters addresses attacks that attempt to solve the underlying mathematical problem, but it does not authenticate the exchanged public values. Certificates, signatures, or authenticated pre-shared keys are needed to bind the key exchange to the intended parties.

QUESTION NO: 3

(How does adding salt to a password improve security?)

A. Salt creates a different hash if two people use the same password.

B. Salt enforces the complexity rules for passwords.

C. Salt ensures two people do not have the same password.

D. Salt prevents users from reusing the same password.

ANSWER: A

Explanation:

Salt creates a different hash if two people use the same password. A salt is a randomly generated value that is unique for each password record and is combined with the password before a password-hashing function derives the stored verifier. Thus, even when two users select identical passwords, the differing salts cause the resulting stored password hashes to differ. The salt is normally stored with the hash; it does not need to be secret to provide this benefit.

Using a unique salt per password prevents an attacker who obtains a password database from immediately identifying accounts with the same password simply by matching identical hash values. It also makes precomputed hash-lookup resources, including rainbow tables, ineffective across the database because the attacker must perform work separately for each salt. These protections are most effective when salts are generated using a cryptographically secure random-number generator and password hashes are produced with an adaptive password-hashing algorithm such as Argon2id, bcrypt, PBKDF2, or scrypt. NIST states that verifiers should use a salt that is at least 32 bits and chosen arbitrarily for each stored password; modern implementations commonly use substantially longer random salts. See NIST SP 800-63B and the OWASP Password Storage Cheat Sheet.

QUESTION NO: 4

(Which encryption algorithm uses an 80-bit key and operates on 64-bit data blocks?)

A. Twofish

B. Blowfish

C. Camellia

D. Skipjack

ANSWER: D

Explanation:

Skipjack is correct because it was specifically designed as a symmetric block cipher with a fixed 80-bit key and a 64-bit block size. In a block cipher, the block size is the fixed amount of plaintext processed by one encryption operation; Skipjack therefore encrypts data in 64-bit units while using the same 80-bit secret key for encryption and decryption. The algorithm was developed for the U.S. government's Clipper-chip initiative and published by NIST in FIPS PUB 185. Its design uses an unbalanced Feistel-network structure and performs 32 rounds, but the identifying parameters relevant to this question are its 80-bit key and 64-bit block. NIST's specification explicitly states that Skipjack encrypts and decrypts 64-bit data blocks under control of an 80-bit key. Although Skipjack is historically important, its 80-bit key length no longer provides an appropriate security margin for modern long-term protection; contemporary systems generally use algorithms and key lengths recommended in current cryptographic guidance. The historical algorithm specification is available in NIST FIPS PUB 185: Skipjack, and NIST's current guidance on selecting stronger cryptographic key sizes is available in NIST SP 800-57 Part 1.

QUESTION NO: 5

(How often are transactions added to a blockchain?)

A. Approximately every 10 minutes

B. Approximately every 30 minutes

C. Approximately every 1 hour

D. Approximately every 24 hours

ANSWER: A

Explanation:

Approximately every 10 minutes is correct in the Bitcoin context. Bitcoin transactions are broadcast to the peer-to-peer network and held in nodes' mempools until a miner includes a set of them in a newly mined block. Once that block is accepted by the network, those transactions are recorded on the blockchain. Bitcoin's proof-of-work protocol is designed to target one new block approximately every 10 minutes on average, rather than guaranteeing that each individual block will take exactly 10 minutes. Mining is probabilistic, so a particular block may be discovered sooner or later. To preserve the long-term average as total mining power changes, Bitcoin readjusts the proof-of-work difficulty every 2,016 blocks. The Bitcoin Developer Guide describes blocks as collections of transactions and explains the role of proof of work and difficulty in block creation. Bitcoin Core's developer documentation also identifies the expected block interval as 10 minutes. Although other blockchain networks use different consensus mechanisms and block intervals, the stated timing is the conventional answer when this question refers generally to blockchain transaction addition in the standard Bitcoin-based instructional context. See the Bitcoin Developer Guide's blockchain overview and Bitcoin's block-chain reference documentation.

QUESTION NO: 6

(What is an attribute of RC4 when used with WEP?)

A. 40-bit key

B. 128-bit key

C. 256-bit key

D. 512-bit key

ANSWER: A

Explanation:

40-bit key is correct because the original Wired Equivalent Privacy implementation commonly combined a 40-bit shared WEP secret key with a 24-bit initialization vector for each transmitted frame. That combined value supplied the keying material used by the RC4 stream cipher, and it was often marketed as "64-bit WEP" because it counted both the secret portion and the transmitted initialization vector. The actual secret key portion, however, is 40 bits. This is the characteristic normally being tested when a question refers to RC4 as used with original WEP.

The initialization vector changes on a per-packet basis and is sent in the clear, so it is not equivalent to 40 additional secret bits. WEP later also supported an extended 104-bit secret-key configuration, frequently marketed as 128-bit WEP after including the same 24-bit initialization vector. Nevertheless, 40-bit WEP is the standard legacy configuration identified by this question. RFC 3580 describes WEP's 40-bit secret key and 24-bit initialization vector construction: RFC 3580. NIST's wireless-security guidance also documents WEP's legacy keying approach and its security limitations: NIST SP 800-48.

QUESTION NO: 7

(Which cipher uses shifting letters of the alphabet for encryption?)

A. SHA-1

B. Vigenère

C. Bifid

D. Caesar

ANSWER: D

Explanation:

The Caesar cipher is correct because it encrypts alphabetic text by shifting every letter a fixed number of positions through the alphabet. For example, using a shift of three positions transforms A into D, B into E, and so on; the alphabet wraps around at the end, so X becomes A. The same fixed displacement is applied consistently to every character in the message, which is why the Caesar cipher is also commonly called a shift cipher. Decryption simply reverses that same displacement. This cipher is historically important as a foundational example of substitution encryption, although it is not secure for modern use because its small number of possible shifts makes exhaustive testing practical. The Caesar cipher is described by Britannica as a simple substitution cipher in which each plaintext letter is replaced by a letter a fixed distance away in the alphabet: Britannica: Caesar cipher. The National Institute of Standards and Technology's Cryptographic Standards and Guidelines resources provide broader context on why modern cryptography relies on substantially stronger, rigorously designed cryptographic algorithms: NIST Cryptographic Standards and Guidelines.

QUESTION NO: 8

(What type of encryption uses different keys to encrypt and decrypt the message?)

A. Symmetric

B. Private key

C. Secure

D. Asymmetric

ANSWER: D

Explanation:

Asymmetric encryption is the correct answer because it uses a related pair of cryptographic keys rather than one shared secret. For confidentiality, a sender encrypts data with the recipient's public key, and the recipient decrypts it with the corresponding private key. The keys are mathematically linked so that an operation performed with one key can be reversed only with its paired key; possession of the public key does not reveal the private key. This public/private-key model lets people distribute a public encryption key openly while retaining exclusive control over the private decryption key. It is therefore useful when parties need to establish secure communication without having previously exchanged the same secret key through a protected channel. Public-key cryptography is also the basis for common systems such as RSA and elliptic-curve cryptography, as well as for secure key exchange and digital-signature workflows. NIST describes public-key cryptography as using a key pair composed of a public key and a private key, and explains the distinct roles those keys can have in cryptographic operations. See the NIST definition of public-key cryptography and NIST's Digital Signature Standard for authoritative background on public/private key pairs.

QUESTION NO: 9

(What is the relationship between Secure Sockets Layer (SSL) and Transport Layer Security (TLS)?)

A. SSL is the replacement of TLS.

B. SSL and TLS are identical in function and security.

C. SSL is only used in email encryption.

D. TLS replaced SSL to provide improved security.

ANSWER: D

Explanation:

TLS replaced SSL to provide improved security. SSL was the original protocol family used to protect application communications, such as web connections, by providing confidentiality through encryption, integrity protections, and authentication capabilities. TLS was developed as SSL's standardized successor: TLS 1.0 was derived from SSL 3.0 but made protocol improvements, and subsequent TLS releases further modernized the permitted cryptography and handshake design. Today, TLS is the protocol used to secure HTTPS and many other client-server services. Although "SSL" is still

frequently used informally in product names and phrases such as “SSL certificate,” current secure deployments use TLS rather than the obsolete SSL protocol versions. SSL 2.0 and SSL 3.0 have known design weaknesses and are deprecated; modern guidance is to disable them and use current TLS versions, preferably TLS 1.3 where supported. The Internet Engineering Task Force specifies TLS 1.3 in RFC 8446, which notes that the protocol was designed to improve security over prior versions. The National Institute of Standards and Technology likewise provides implementation guidance for TLS and recommends secure, modern configurations. See RFC 8446: The Transport Layer Security (TLS) Protocol Version 1.3 and NIST SP 800-52 Rev. 2.

QUESTION NO: 10

(How is Public Key Infrastructure (PKI) commonly utilized in web browsers?)

A. To compress encrypted messages for storage

B. To authenticate users during data transmission

C. To securely manage digital certificates and keys

D. To encrypt data at rest

ANSWER: C

Explanation:

To securely manage digital certificates and keys is correct because web browsers use PKI as the trust framework behind HTTPS/TLS connections. A website presents an X.509 digital certificate containing its public key and identity information, including the domain names for which it is valid. The browser evaluates that certificate’s validity period, domain-name binding, signature, and chain of trust through intermediate certificate authorities to a trusted root certificate already present in its trust store. This validation allows the browser to establish that it is communicating with the intended website rather than an impersonator.

After the site’s certificate has been validated, the browser and server use TLS to authenticate the server and securely establish session keys. Those symmetric session keys then protect the confidentiality and integrity of the web traffic for that connection. Thus, PKI’s browser role includes handling certificates, public/private key relationships, trusted certificate authorities, and certificate-validation policies that make authenticated secure web sessions possible. The CA/Browser Forum’s Baseline Requirements describe the certificate ecosystem used for publicly trusted TLS certificates, and Mozilla documents how its browser trust store governs which certificate authorities Firefox trusts in its Root Store Policy.

QUESTION NO: 11

(A developer converts a customer identifier into Base64 text before storing it in a database. An unauthorized database user can read the Base64 value. What security property does Base64 provide for the identifier?)

A. Confidentiality, because decoding requires a private key

B. Integrity, because any change causes decoding to fail

C. No cryptographic confidentiality or integrity protection

D. Nonrepudiation, because the value identifies a customer

ANSWER: C

Explanation:

Base64 provides no confidentiality for the identifier. It is an encoding scheme that represents binary data using printable characters and can be reversed without a secret key. Anyone who has the encoded value can decode it using commonly available tools. Encryption would be required to protect confidentiality, while a hash could support integrity checking or password verification in an appropriate design but would not allow normal recovery of the original identifier.