Patrick Kevorkian
HW 8
Network Security

Fall 2017

1.1

a) Extensible Authentication Protocol or EAP is an authentication framework used in wireless networks. A user requests access through an access point. The access point requests an ID from the user and sends it to an authentication server. This server validates the validity of the ID. After the access point gets the verification from the user and sends it back to the authentication server, the user is connected to the wireless network.

b) 1. EAP Pre Shared Key (EAP-PSK): Way of mutual authentication and session key derivation using a pre shared key. Provides a protected communication channel when mutual authentication is successful for both parties to communicate over.

2. EAP Password (EAP-PWD): Uses a shared password for authentication. The password may be low a entropy one and may be drawn from some set of possible passwords, like a dictionary, which is available to an attacker. The underlying key exchange is resistant to active attack, passive attack, and dictionary attack.

3. EAP Encrypted Key Exchange (EAP-EKE): This document defines an authentication mechanism for EAP called EAP-EKE based on the Encrypted Key Exchange (EKE) protocol. This method provides mutual authentication through the use of a short, easy to remember password. Compared with other common authentication methods, EAP-EKE is not susceptible to dictionary attacks. Neither does it require the availability of public-key certificates.

4. EAP-IKEv2: Extensible Authentication Protocol (EAP) method that is based on the Internet Key Exchange (IKEv2) protocol. EAP-IKEv2 provides mutual authentication and session key establishment between an EAP peer and an EAP server. It supports authentication techniques that are based on passwords, high-entropy shared keys, and public key certificates. EAP-IKEv2 further provides support for cryptographic ciphersuite negotiation, hash function agility, identity confidentiality (in certain modes of operation), fragmentation, and an optional "fast reconnect" mode.

c) Extensible Authentication Protocol over LAN: The same three main components are defined in EAP and EAPoL to accomplish the authentication conversation. First a supplicant (Port Authentication Entity (PAE) seeking access to network resources), Second Authenticator (PAE that controls network access), and Last, Authentication Server (a RADIUS/AAA server)

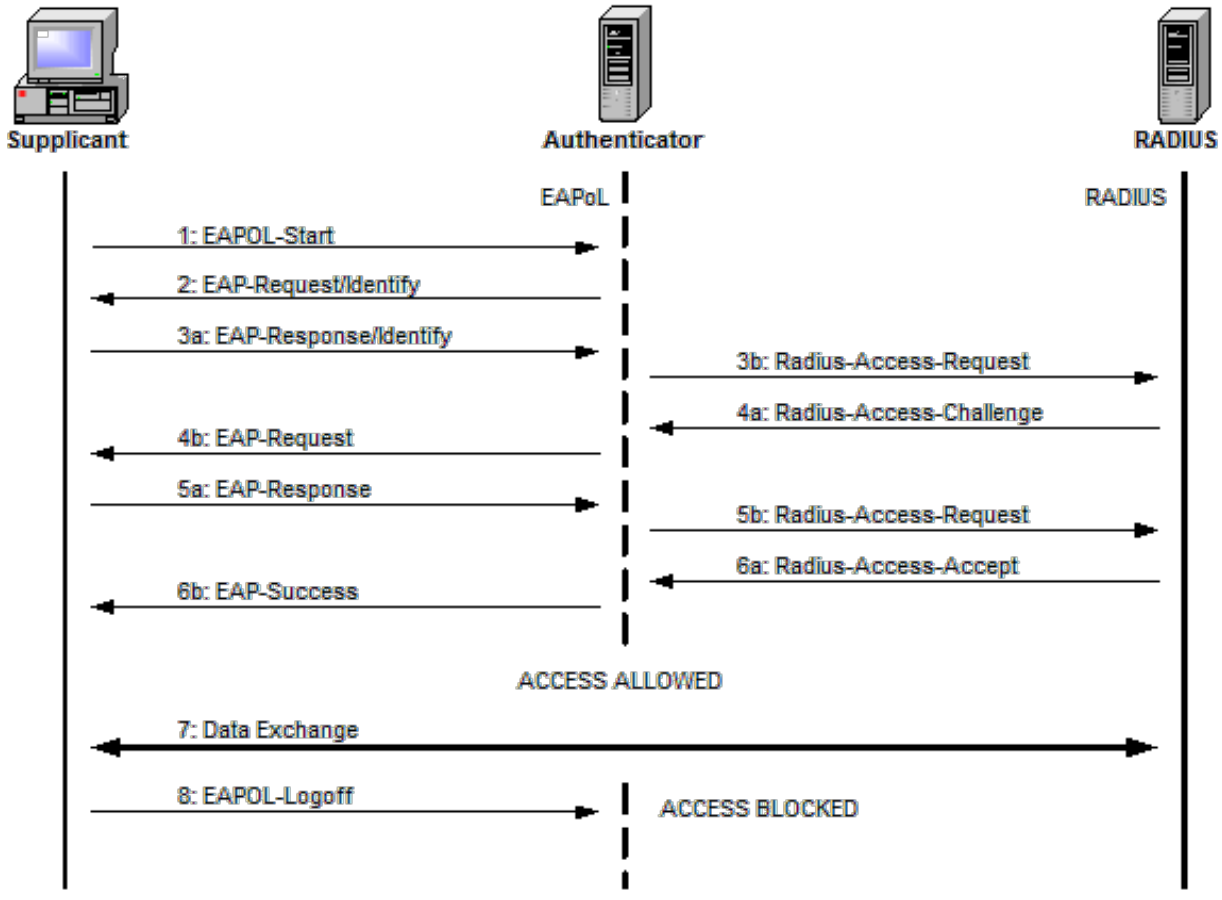


Figure 2: Sample EAPoL Exchange

d) Is a Standard for port-based Network Access Control (PNAC). It is part of the IEEE 802.1 group of networking protocols. It provides an authentication mechanism to devices wishing to attach to a LAN or WLAN. IEEE 802.1X defines the encapsulation of the Extensible Authentication Protocol (EAP) over IEEE 802,[1][2] which is known as "EAP over LAN" or EAPOL 802.1X authentication involves three parties: a supplicant, an authenticator, and an authentication server. The supplicant is a client device (such as a laptop) that wishes to attach to the LAN/WLAN. The term 'supplicant' is also used interchangeably to refer to the software running on the client that provides credentials to the authenticator. The authenticator is a network device, such as an Ethernet switch or wireless access point; and the authentication server is typically a host running software supporting the RADIUS and EAP protocols. In some cases, the authentication server software may be running on the authenticator hardware.

1.2

Layers:

a) Layer 1: “Method” - The textbook essentially refers to this as the discovery stage which in itself has 3 stages. First the “Network and security capability discovery,” here the STAs discover the existence of a network in which to communicate. This is sone in several ways: Either the AP will broadcast is security capabilities by RSN IE in a specific channel through a bacon frame, or weapons to stations probe request through a probe response frame. A wireless station may discover available access points and corresponding security capabilities by either passively monitoring the Beacon frames or actively probing every channel. Next “Open system

Authentication" which maintains backwards compatibility with the IEEE 802.11 state machine. The two devices STA and AP change identifiers. Last is "Association," where they agree on the on a set of security capabilities to be used. STA sends an Association Request frame to the AP. In this frame, the STA specifies one set of matching capabilities (one authentication and key management suite, one pairwise cipher suite, and one group-key cipher suite) from among those advertised by the AP. If there is no match in capabilities between the AP and the STA, the AP refuses the Association Request. The STA blocks it too, in case it has associated with a rogue AP or someone is inserting frames illicitly on its channel. Controlled ports are blocked, and no user traffic goes beyond the AP.

b) Layer 2: - Consists of 3 parts: An access point or NAS that requires EAP authentication prior to granting access to a network (an EAP authenticator). Client computer that is attempting to access a network. (an EAP peer). Last A server computer that negotiates the use of a specific EAP method with an EAP peer, validates the EAP peer's credentials, and authorizes access to the network. Typically, the authentication server is a Remote Authentication Dial-In User Service (RADIUS) server (an Authentication server). The authentication server functions as a backend server that can authenticate peers as a service to a number of EAP authenticators. The EAP authenticator then makes the decision of whether to grant access. This is referred to as the EAP pass- through mode. Less commonly, the authenticator takes over the role of the EAP server; that is, only two parties are involved in the EAP execution.

c) Layer 3: EAP Layer- As a first step, a lower-level protocol, such as PPP (point-to-point protocol) or IEEE 802.1X, is used to connect to the EAP authenticator. The software entity in the EAP peer that operates at this level is referred to as the supplicant. EAP messages containing the appropriate information for a chosen EAP method are then exchanged between the EAP peer and the authentication server. EAP messages may include the following fields: Code: Identifies the Type of EAP message. The codes are Request (1), Response (2), Success (3), and Failure Identifier: Used to match Responses with Requests. Length: Indicates the length, in octets, of the EAP message, including Code, Identifier, Length, and Data fields. Data: Contains information related to authentication. Typically, the Data field consists of a Type subfield, indicating the type of data carried, and a Type- Data field.

d) Layer 4: Lower Layer- One protocol used for this purpose is IEEE 802.1X, discussed in the next section. The first pair of EAP Request and Response messages is of Type identity, in which the authenticator requests the peer's identity, and the peer returns its claimed identity in the Response message. This Response is passed through the authenticator to the authentication server. Subsequent EAP messages are exchanged between the peer and the authentication server. Upon receiving the identity Response message from the peer, the server selects an EAP method and sends the first EAP message with a Type field related to an authentication method. If the peer supports and accepts the selected EAP method, it replies with the corresponding Response message of the same type. Otherwise, the peer sends a NAK, and the EAP server either selects another EAP method or

aborts the EAP execution with a failure message. The selected EAP method determines the number of Request/Response pairs. During the exchange the appropriate authentication information, including key material, is exchanged. The exchange ends when the server determines that authentication has succeeded or that no further attempt can be made and authentication has failed.

2.1

a) A SSL connection is a transport that provides a suitable type of service. In the case of SSL, such a connection is peer-to-peer and the connections are transient. Furthermore, each connection is associated with one SSL session, which is defined as the association between a client a server. A session is created by the Handshake Protocol, and it defines a set of cryptographic security parameters which can be shared among multiple connections.

b)

SSL Session state parameters:

1. Session identifier: This is an arbitrary byte sequence by the server to identify an active or resumable session state.

2. Peer certificate: This is an X509.v3 certificate of the peer. This element of the state may be null.

3. Compression method: This is the algorithm used to compress data prior to the encryption.

4. Cipher spec: This specifies the bulk data encryption (null, AES, etc.) and a hash algorithm (MD5 or SHA-1, etc.) sued for MAC calculation. This also defines cryptographic attributes such as the hash_size.
5. Master secret: This is a 48-byte secret shared between the client and the server.
6. Is resumable: This is a flag indicating whether the session can be used to initiate new connections.

SSL Session Connection parameters:
1. Server and client random: These are byte sequences that are chosen by the server and client for each connection
2. Server write MAC secret: This is the secret key used in MAC operations on data sent by the server.
3. Client write MAC secret: This is the secret key used in MAC operations on data sent by the client.
4. Server write key: This is the secret encryption key for data encrypted by the server and decrypted by the client.
5. Client write key: This is the symmetric encryption key for data encrypted by the client and decrypted by the server.
6. Initialization vectors: When a block cipher in CBC mode is used, an initialization vector (IV) is maintained for each key. This field is first initialized
by the SSL Handshake Protocol. Thereafter, the final ciphertext block from each record is preserved for use as the IV with the following record.

7. Sequence numbers: Each party maintains separate sequence numbers for transmitted and received messages for each connection. When a party sends or receives a change cipher spec message, the appropriate sequence number is set to zero. Sequence numbers may not exceed 2^64-1.

b) Hyper Text Transfer Protocol Secure (HTTPS) is the secure version of HTTP, the protocol over which data is sent between your browser and the website that you are connected to. The 'S' at the end of HTTPS stands for 'Secure'. It means all communications between your browser and the website are encrypted. HTTPS is often used to protect highly confidential online transactions like online banking and online shopping order forms. HTTPS pages typically use one of two secure protocols to encrypt communications - SSL (Secure Sockets Layer) or TLS (Transport Layer Security). Both the TLS and SSL protocols use what is known as an 'asymmetric' Public Key Infrastructure (PKI) system. An asymmetric system uses two 'keys' to encrypt communications, a 'public' key and a 'private' key. Anything encrypted with the public key can only be decrypted by the private key and vice-versa.

c) The initial version, SSH1 was focused on providing a secure remote logon facility to replace TELNET and other remote logon schemes that provided no security. SSH also provides a more general client/server capability and can be used for such network functions as file transfer and e-mail.

2.2

a) SSL uses symmetric one-time session keys and it has the capability to negotiate a stronger cipher to be used during session.
b) SSL uses per session random numbers (of client and server) to generate the session key. It helps in randomizing the cipher text.
c) The random numbers used in each session has the first 4 bytes as the time stamp, so they are different for each session.
d) Mutual authentication with certificates.
e) Passwords are encrypted.
f) SSL does not use IP addresses to authenticate the client and server.
g) If the attacker hijacks the connection after authentication, he has no way of knowing the encryption key. Therefore, the Alert protocol will detect if the attacker tries to send data as a legitimate user and closes the connection eventually. Even if the attacker hijacks it during handshaking, the attacker does not know the password and hence cannot succeed during the password authentication phase.
h) Can not be defeated. SSL is not stateless and working on top of TCP.

3.1

a) IEEE 802.11i addresses three security areas:

1) authentication,

2) key management, and

3) data transfer privacy.

b)

1.Discovery Phase: The purpose of this phase is for an (STA) and an access point (AP) to recognize each other, agree on a set of security capabilities, and establish an association for future communication using those security capabilities.

2.Authentication Phase: This phase enables mutual authentication between a wireless station (STA) and an authenticated server (AS) located in the distributed system (DS). Authentication is designed to allow only authorized stations to use the network and to provide the STA with assurance that it is communicating with a legitimate network.

3.Key Management Phase: During the key management phase, a variety of cryptographic keys are generated and distributed to wireless stations (STAs). There are two types of keys: pairwise keys (used for communication between a STA and an access point (AP)) and group keys (used for multi-cast communication). Both the pairwise and group key have different hierarchies.

4.Protected Data Transfer Phase: During this phase, the IEEE 802.11i defines two schemes for protecting data transmitted in 802.11 MPDUs: the Temporal Key Integrity Protocol (TKIP), and the Counter Mode-CBC MAC Protocol (CCMP).

c)

The TKIP is designed to require only software changes to devices that are implemented with the older wireless LAN security approach called the Wire Equivalent Privacy (WEP). The CCMP is intended for the newer IEEE 802.11 devices that are equipped

with the hardware to support this scheme. They both support two services: 1) message integrity and 2) data confidentiality.

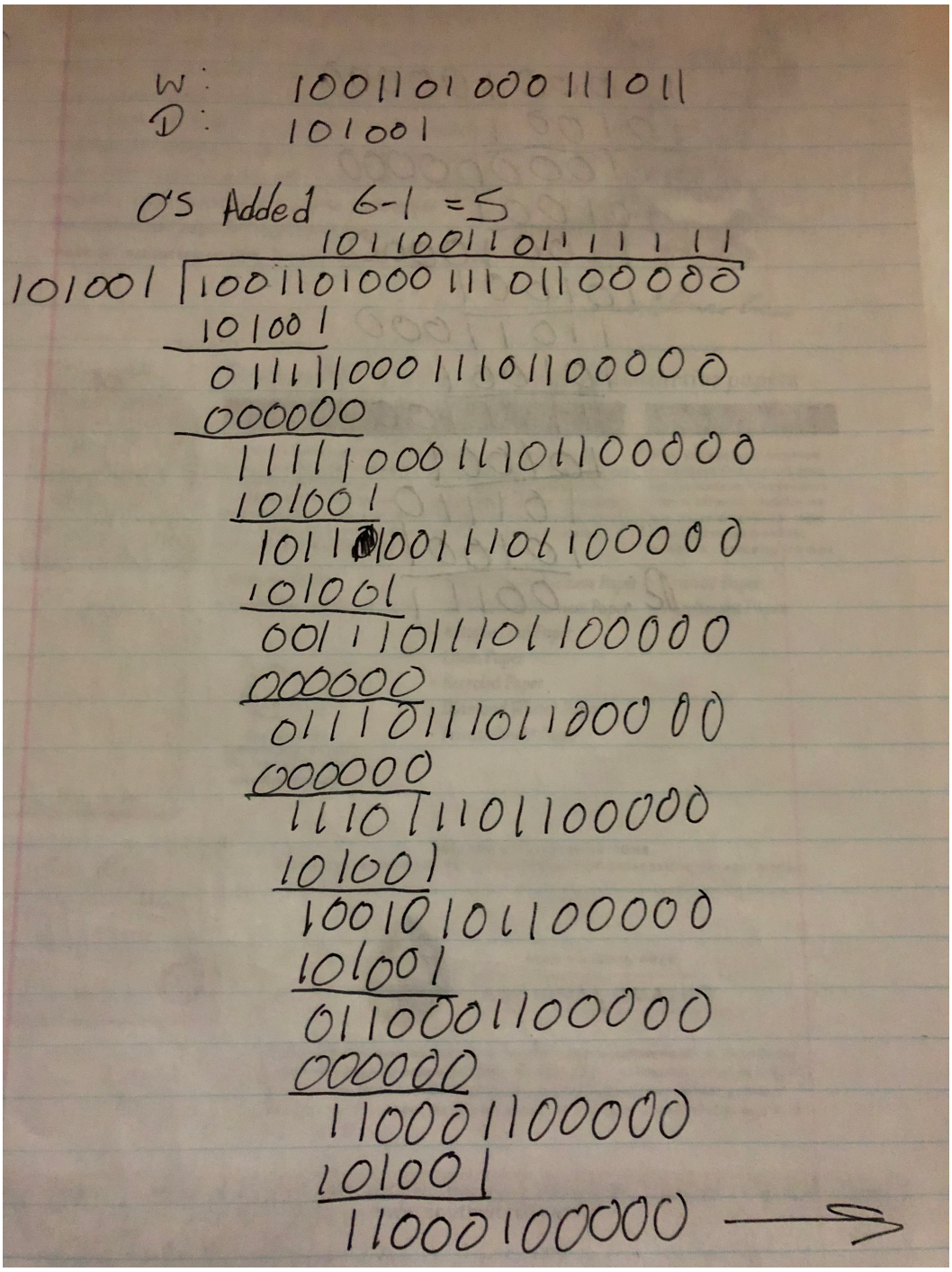
For message integrity, TKIP uses a message integrity code (MIC) to the 802.11 MAC frame after the data field, and CCMP uses the cipher block chaining message authentication code
(CBC-MAC). For data confidentiality, the TKIP also uses RC4 for encryption, while the CCMP uses CTR block cipher mode of operation with AES for encryption.

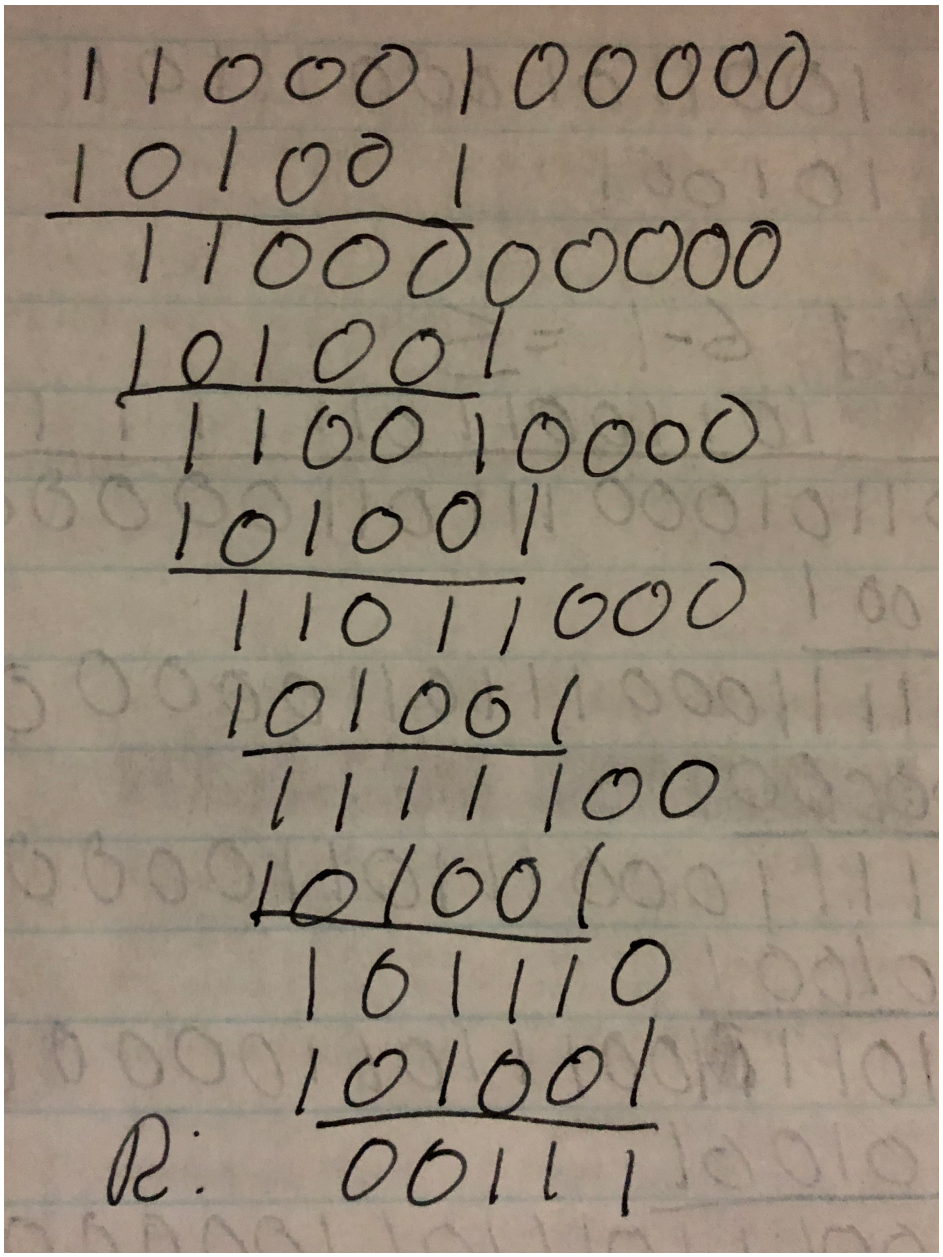
3.2

Cyclic redundancy checking is a method of checking for errors in data that has been transmitted on a communications link. A sending device applies a 16- or 32-bit polynomial to a block of data that is to be transmitted and appends the resulting cyclic redundancy code (CRC) to the block. The receiving end applies the same polynomial to the data and compares its result with the result appended by the sender. If they agree, the data has been received successfully. If not, the sender can be notified to resend the block of data. The ITU-TS (CCITT) has a standard for a 16-bit polynomial to be used to obtain the cyclic redundancy code (CRC) that is appended. IBM's Synchronous Data Link Control and other protocols use CRC-16, another 16-bit polynomial. A 16-bit cyclic redundancy code detects all single and double-bit errors and ensures detection of 99.998% of all possible errors. This level of detection assurance is considered sufficient for data transmission blocks of 4 kilobytes or less. For larger transmissions, a

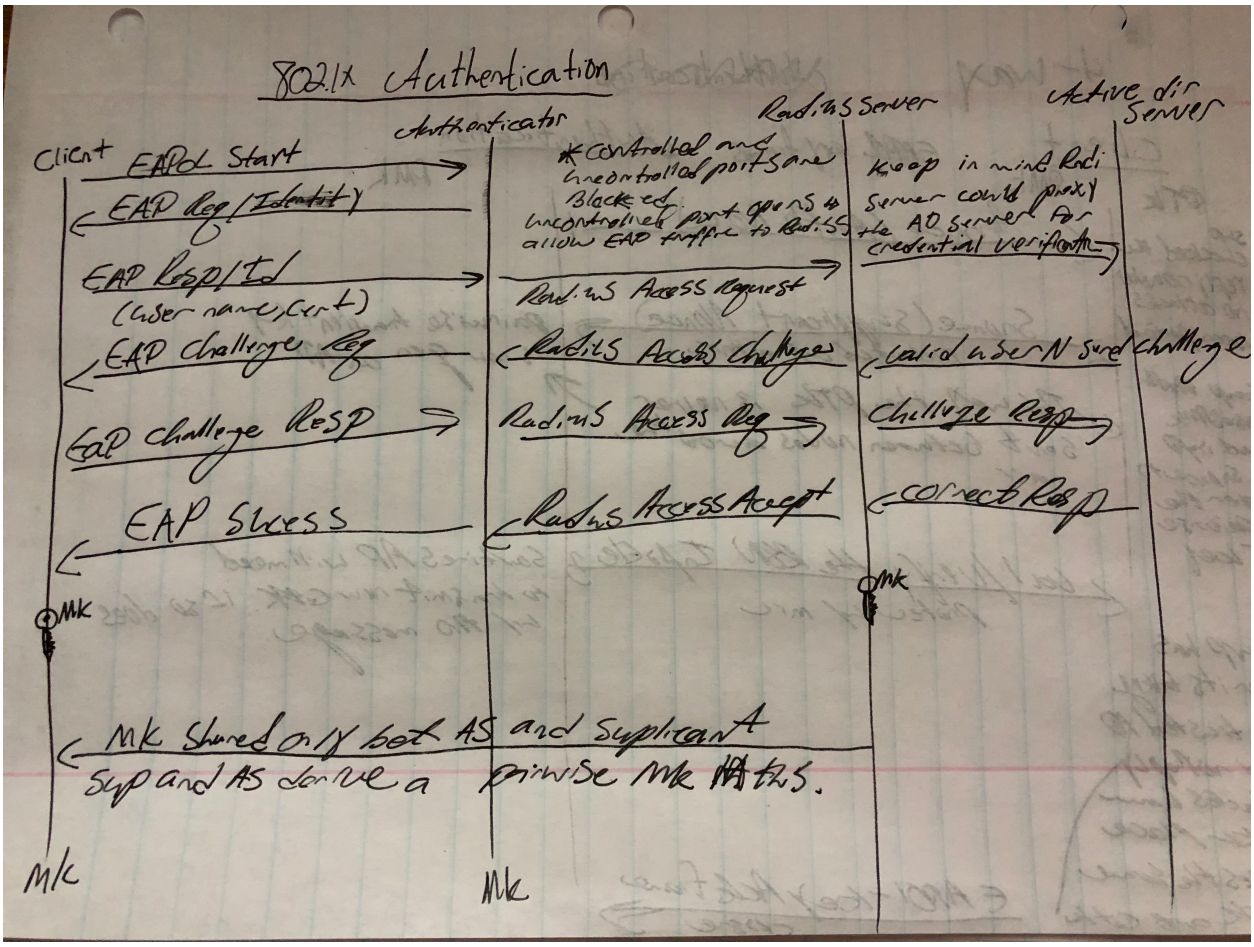
32-bit CRC is used. The Ethernet and token ring local area network protocols both used a 32-bit CRC.

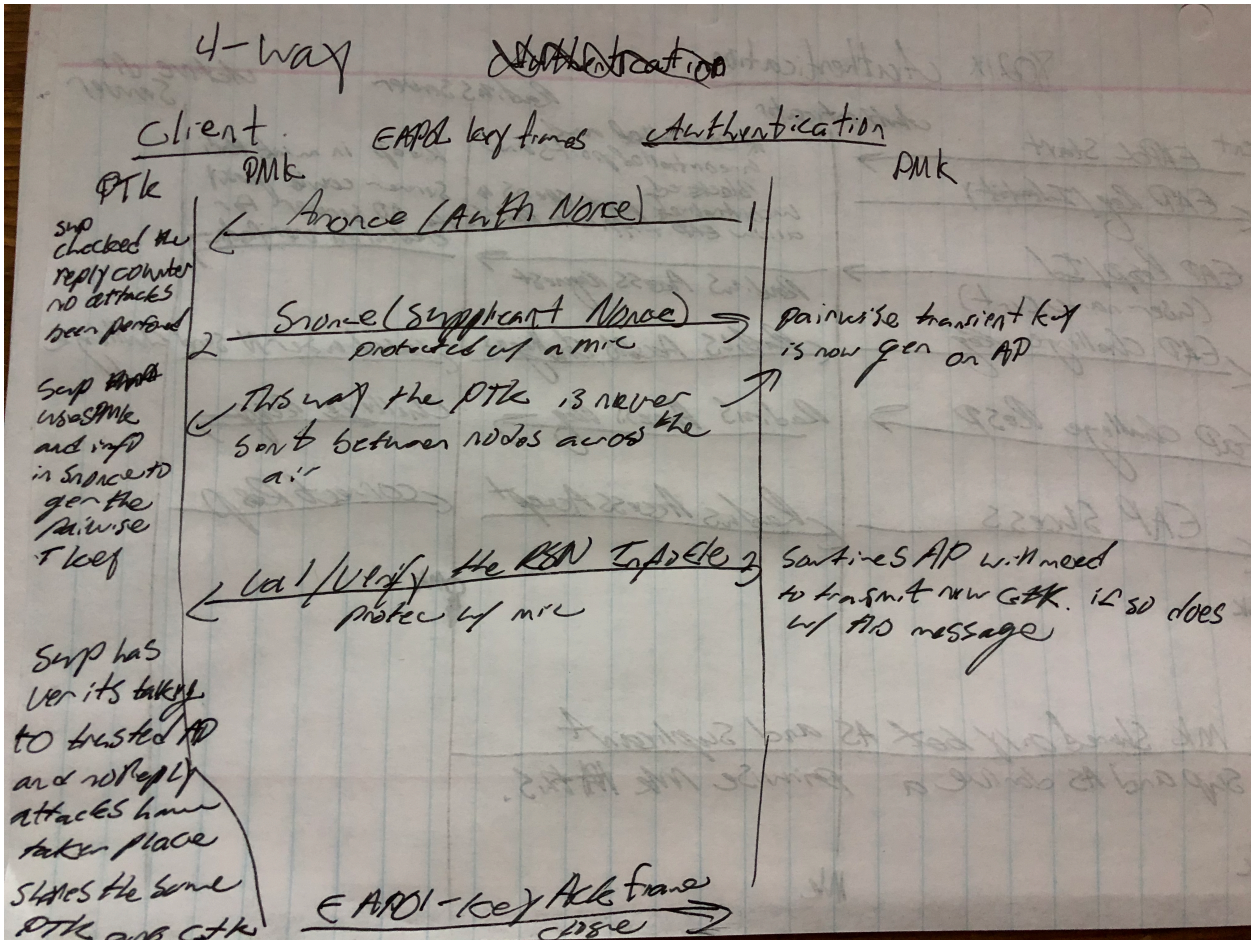
Given the generator polynomial x^5 + x^3 + 1 (CRC-5-EPC for Gen2 RFID), and the message 0x9A3B, what is the CRC checksum?

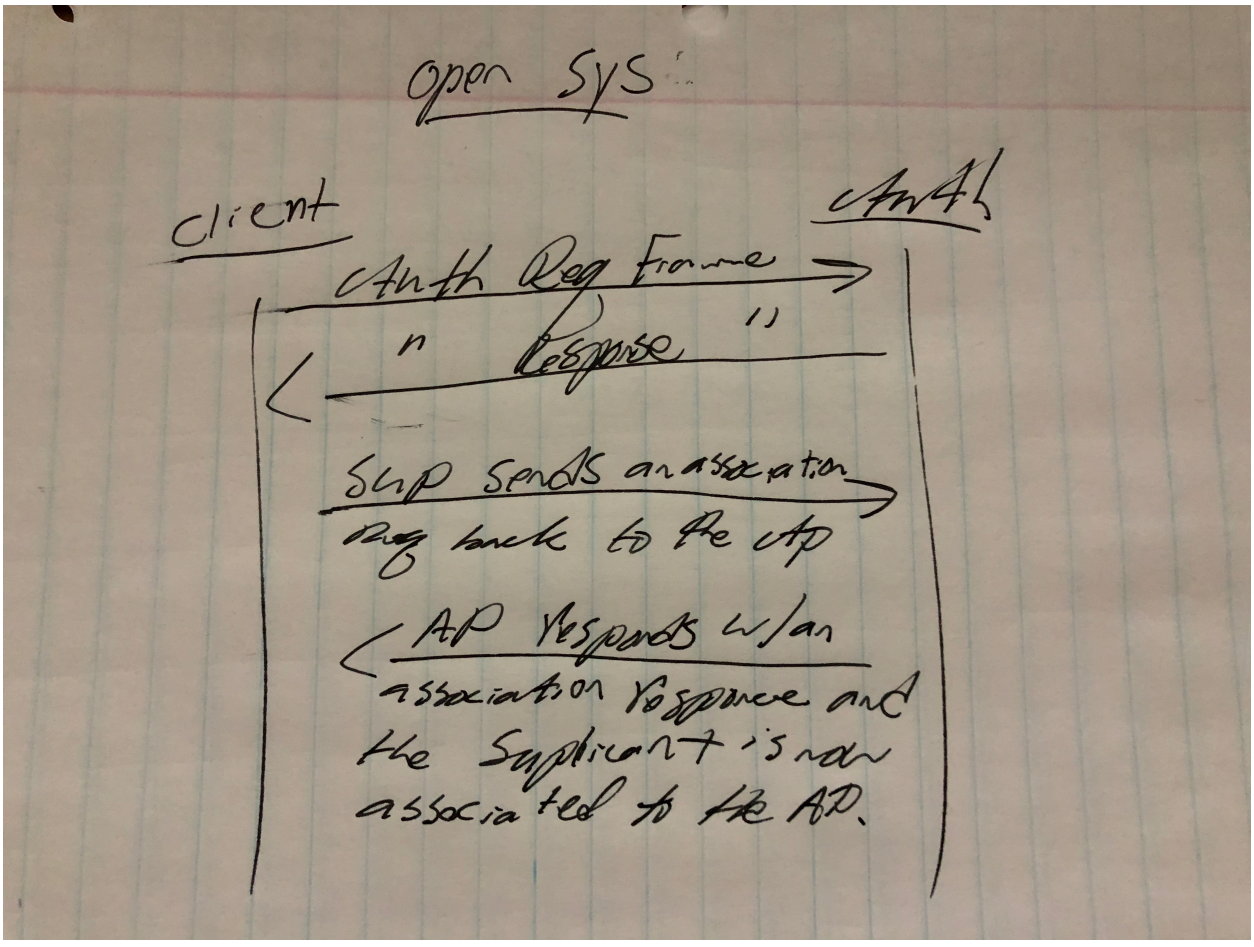



11000100000
101001
11000000000
101001
1100010000
101001
110111000
10100 1
11111100
101001
10111 0
101001
R: 00111

3.3


802.1x Authentication
Client
Authenticator
Radius Server
Active dir Server
EAPol Start
EAP Req/Identity
EAP Resp/Id
(user name, Cert)
Radius Access Request
EAP challenge Req
Radius Access Challenge
valid user N sure challenge
EAP challenge Resp
Radius Access Req
Challenge Resp
EAP Success
Radius Access Accept
correct Resp
MK
MK
MK shared only bet AS and Supplicant
Supp and AS derive a
pairwise MK
MK
MK


4-way
Client
EAPOL key frames
Authentication
PMK
PTK
PMK
Anonce (Auth Nonce)
1
Snonce (Supplicant Nonce)
protected w/ a mic
2
pairwise transient key is now gen on AP
This way the PTK is never sent between nodes across the air
Cal/Verify the RSN IE/Cipher
protect w/ mic
3
Confirms AP w/ need to transmit new GTK. If so does w/ this message
Supp has ver its taking to trusted AP and no Repl attacks have taken place
EAPOL-Key Ack frame
close


Open Sys
client
Auth
Auth Req Frame
1)
n
Response
Supp sends an association Req back to the AP
AP responds w/ an association Response and the Supplicant is now associated to the AP.

3.4

a) As AP (Access Point) has compatibility of remembering previous sent random numbers, AP can verify or check whether the results resotesd was encrypted with the correct key or do not.

In order to encrypt the random value the STA (Station) Must know the key.

b) The WEP (Wired Equivalent privacy) authentication scheme missing to state key of AP (Access Point) to STA (Station)

Authentication here is in only one way that is STA (Station) sends requests for authentication, AP (Access Point) sends message of 128 bit random number. Since here the key is known only by the AP (Access Point), authentication is only one way.

c) The cryptographic weakness of the WEP Authentication is, if an attacker monitors then this WEP (Wired Equivalent Privacy) Authentication provides plaintext-ciphertext pair to be used for cryptanalysis.

4.1

a) Authentication (Sign/Verify)

Confidentiality (Encryption/Decryption)

Compression

Email compatibility

Segmentation and Reassembly

b)

1. So that one can store only the uncompressed message together with signature for later verification.

2. Applying the hash function and signature after compression would constrain all PGP implementations to the same version of the compression algorithm as the PGP compression algorithm is not deterministic.

c)

Radix 64 conversion is a scheme that PGP uses to convert the raw 8-bit binary scheme to a stream of printable ASCII characters. This expands a message by 33%, but the session key and signature portions of the message are relatively compact, and the plaintext message has been compressed. It also blindly converts the input stream to radix-64 format regardless of the content, even if the input happens to be ASCII text.

Most email mail systems only permit the use of blocks consisting of ASCII text, so PGP must provide a service (Radix-64, or R64) of converting the raw 8-bit binary stream to a stream of printable ASCII characters. The reason R64 conversion is useful for an email application is because it blindly converts the input stream to radix-64 format regardless of the content, even if the input happens to be ASCII text. In other words, if the message is signed (but not encrypted) and the conversion is applied to the entire block, the output will still be unreadable to the casual observer. This provides a certain degree of confidentiality.

d)

PGP doesn't require Certificate authorities, PGP depends on a ring of trust and only two paths. There may be multiple lines of trust from a fully trusted authority to a certificate.

X.509 requires CA's and the CA is the root of the trust hierarchy. That CA issues certificates for the next level, and those CA's issue certificates for the next level, and so on. The root CA is trusted by everyone.

4.2

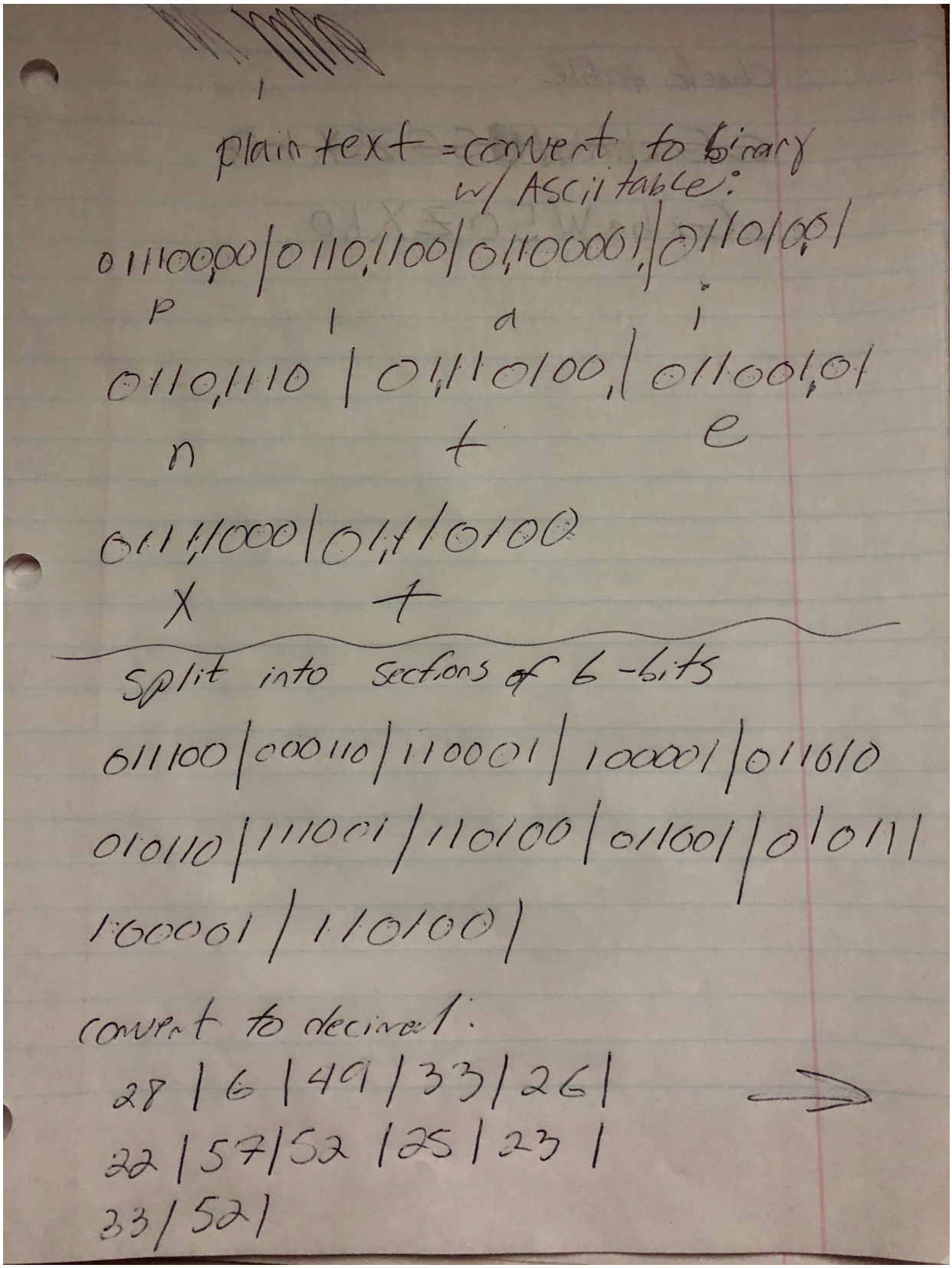
The first 16 bits of the message digest in a PGP signature are translated in the clear. What do you see as issue with respect to security compromise of the hash algorithm? To what extent does it in fact perform its intended function, namely, to help determine if the correct RSA key was used to decrypt the digest?

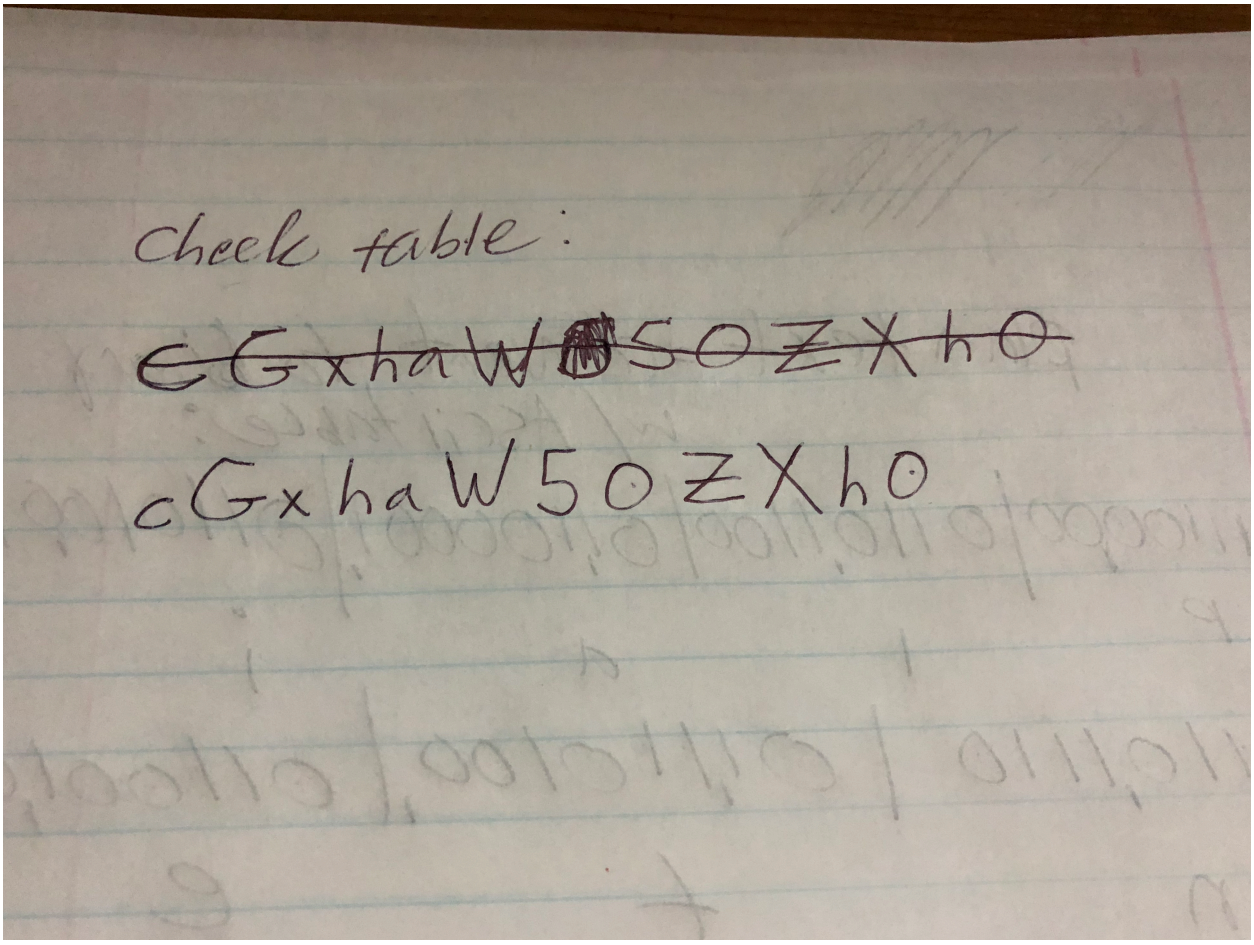
Pretty Good Privacy (PGP)LaRon Walker Master of Information Technology and Internet Pretty Good Privacy (PGP) is a secure method to establish a trust relationship between message senders and recipients. PGP was developed to cover most concerns that are associated with secure message transport, including authentication, confidentiality, compression, and integration flexibility. As discussed in the article Digital Signature (2003), PGP gives its participants the ability to sign each other¶s keys, which establishes a trust relationship between sender and recipient with the help of public keys. Along with this, PGP incorporates a number of modern security components like Rivest-Shamir-Adleman (RSA), Digital Signature Standard(DSS), and Diffie-Hellman for public key encryption along with TripleDES (3DES) covering symmetric block encryption and Secure Hash Algorithm (SHA-1) for hash coding (Stallings,2006). Each of the above techniques has established track records, and has been proven to provide a certain level of security when used alone. When these techniques are used

together,PGP can provide a secure method for message transport addressing most concerns that comes with maintaining message integrity.

one concern that may arise when implementing PGP is that it is commonly used with RSA, which sends the first 16 bits of its signature in plaintext. This information is compressed and encoded in the leading header of the message, and used to verify the intended recipient has the correct key. If the signature information cannot verify the key as being valid, the message isn't sent. Due to this information being sent in plaintext, it may appear as an issue to some security professionals, as it presents the opportunity for hash algorithms to be compromised.Despite this potential drawback, PGP is still considered secure if implemented in conjunction with SHA1 to address this issue. SHA-1 creates hash values of 160 bits, making it very difficult for them to be compromised. This addresses weaknesses discovered in previous secure hashalgorithm versions, and is still considered one of the most secure encryption techniques used today. Using SHA-1 along with RSA can help address concerns associated with information being sent in plaintext. Even though RSA sends 16 bits of its message digest in plaintext, SHA-1can provide another level of security by using message lengths to produce separate 160 bit message digests, which can also be used to verify message integrity.

4.3


plaintext = convert to binary w/ ASCii table:
01110000|01101100|01100001|01101001
p l a i
01101110|01110100|01100101
n t e
01110000|01110100
x t
split into sections of 6-bits
011100|000110|110001|100001|011010
010110|111001|110100|011001|010111
100001|110100|
convert to decimal:
28|6|49|33|26|
22|57|52|25|23|
33|52|


Check table:
~~cGxhaW50ZXNo~~
cGxhaW50ZXNo