Cryptographic Standards and Protocols

An Overview

Prepared by Andrew Colarik, 2014
Lightly edited by Clark Thomborson, 2015-6. V1.21 2017-08-14.

Overview

- Kerberos
- X.509
- X.500
- IPv6
- SSL
- TLS
- IPSec

Kerberos

- Kerberos is a network authentication protocol. It is designed to provide strong authentication for client/server applications by using secret-key cryptography.
- Before a network connection is opened between two entities, Kerberos establishes a shared secret key through a Ticket Granting Server (TGS) that is used for authenticating the parties in the subsequent communications
- Versions of Kerberos also have extensions to utilize public/private keys for authentication
- Versions 4 and 5 (RFC 1510) are in use today
  - v4 has technical deficiencies
    - http://www.isi.edu/div7/publication_files/evolution_of_kerberos.pdf

Category: Authentication

Kerberos

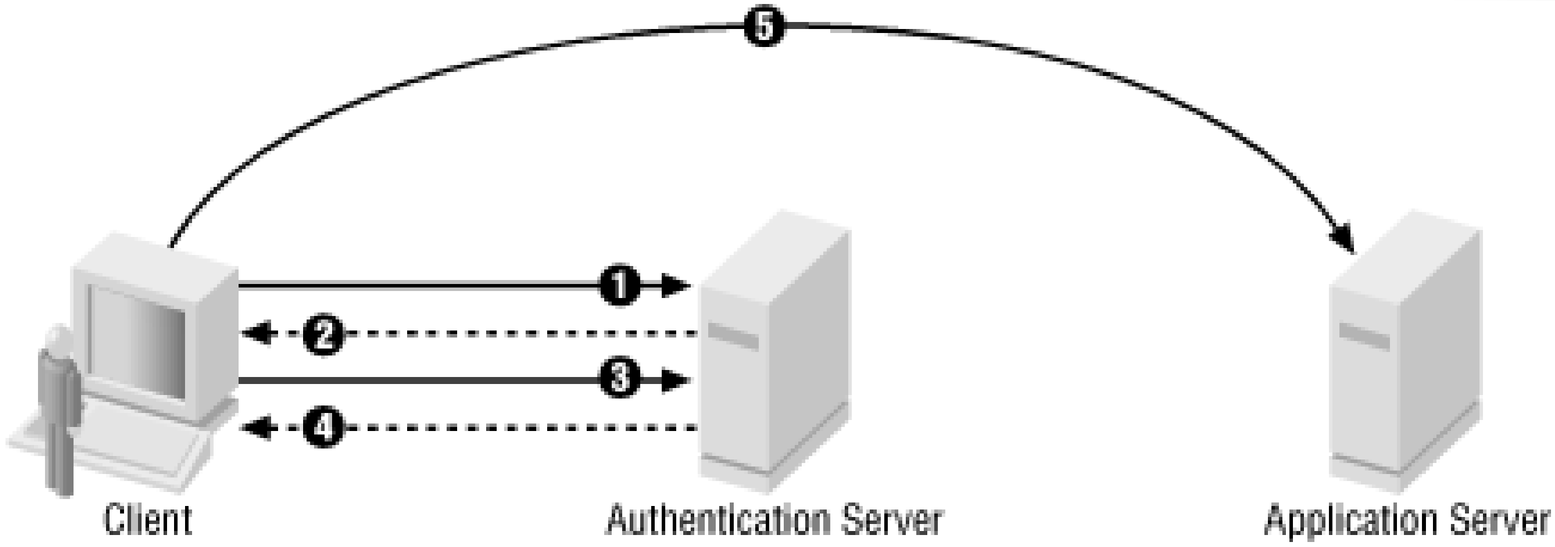
- It provides a centralized private-key third-party authentication in a distributed network
  - Allows users access to services distributed through a network without needing to trust all workstations
  - All trust is handled through a central authentication server
    - Implemented using an authentication protocol based on Needham-Schroeder

Kerberos

- Kerberos environment consists
  - A Kerberos server
  - A number of clients, all registered with the server
  - Application servers, sharing keys with the Kerberos server
    - Termed a realm
  - Typically a single administrative domain
    - If multiple realms, their Kerberos servers must share keys and trust

  - Authentication Server (AS)
    - Users initially negotiate with AS to identify self
    - AS provides a non-corruptible authentication credential
      - Ticket Granting Ticket (TGT)
    - Ticket Granting server (TGS)
      - Users subsequently request access to other services from TGS on basis of users TGT
    - Uses a complex protocol using DES

Kerberos



1. *Request for ticket granting ticket (TGT)*
2. *TGT returned by authentication service*
3. *Request for application ticket (authenticated with TGT)*
4. *Application ticket returned by ticket-granting service*
5. *Request for service (authenticated with application ticket)*

X.509

- To facilitate the identification and security of keys in PKI, a Certificate Authority (CA) is used to authenticate the public key by digitally signing it
  - This is known as a digital certificate
- The validation and invalidation process (authentication) of digital certificates is handled by the Certificate Authority, and is governed by the X.509 de-facto standard.
  - Specifies the semantics of certificates and certificate revocation lists for the Internet PKI

Category: Authentication

X.500

- The X.500 standard is a global directory service that is based on a replicated distributed database
- Programs access the directory services using the X/Open Directory Service (XDS) APIs.
- The XDS API's permit programs to read, compare, update, add, and remove directory entries; list directories; and search for entries based on attributes, while authenticating these activities.
- There are varieties of X.500 products (i.e. Directory Access Protocols) available, and the latest version is LDAP.
  - Lightweight Directory Access Protocol (LDAP) provides the same functions as DAP except it reduces overheads through bypassing much of the session and presentation layers using Distinguished Names (DN)

Category: Authentication

LDAP

- The Lightweight Directory Access Protocol is an open, vendor-neutral, industry standard application protocol for accessing and maintaining distributed directory information services over an Internet Protocol network.
- Directory services play an important role in developing intranet and Internet applications by allowing the sharing of information about users, systems, networks, services, and applications throughout the network.
  - provide any organized set of records
  - often with a hierarchical structure such as a corporate email directory
- A common usage of LDAP is to provide a single-sign-on where one password for a user is shared between many services

http://www.ietf.org/rfc/rfc4511.txt

Category: Authentication

LDAP

- LDAP Data Interchange Format (LDIF)

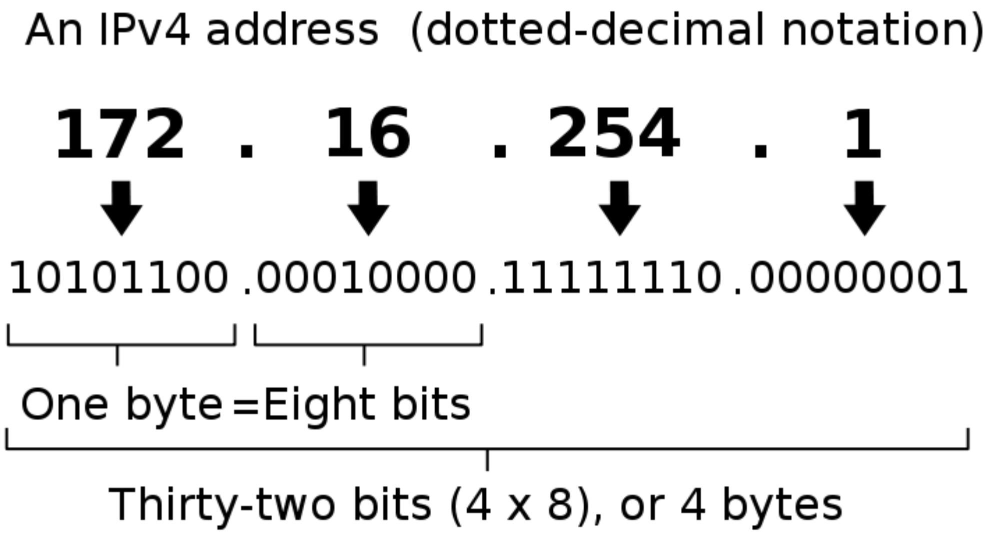
```
dn: cn=John Doe,dc=example,dc=com
cn: John Doe
givenName: John
sn: Doe
telephoneNumber: +1 888 555 6789
telephoneNumber: +1 888 555 1232
mail: john@example.com
manager: cn=Barbara Doe,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
```

IPv6

- The proposed standard Internet Protocol version 6 (IPv6) is the next generation of IP and will eventually replace IPv4.
  - Currently being transitioned throughout the Internet and is backward compatible with version 4.
- IPv6 provides the following added features
  - An increase from the 32-bit address space to 128-bit
  - Provisions for unicast, multicast, and anycast
  - An extension Authentication Header (AH) which provides authentication and integrity (without confidentiality) to IPv6 datagrams
  - An IPv6 Encapsulating Security Header (ESH) which provides integrity and confidentiality to datagrams

http://tools.ietf.org/html/rfc2460

IPv6



An IPv4 address (dotted-decimal notation)

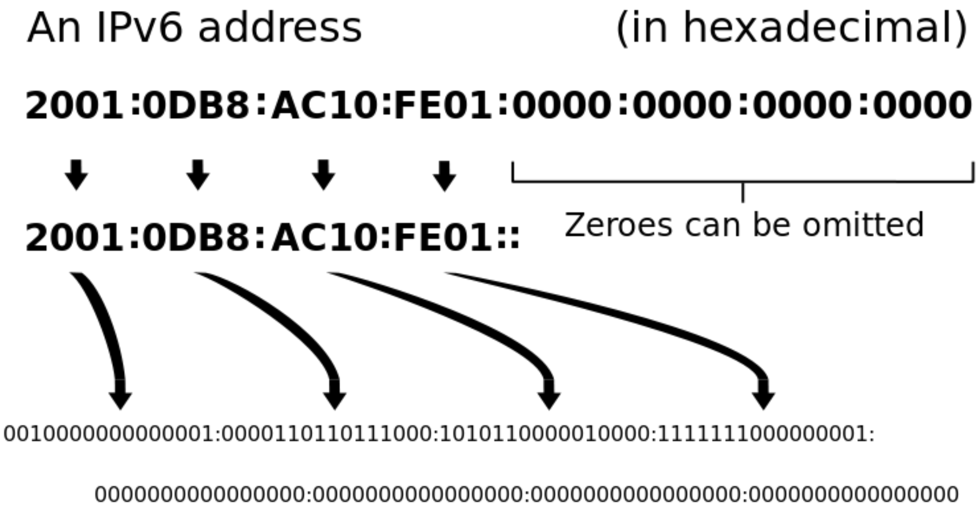
172 . 16 . 254 . 1

10101100.00010000.11111110.00000001

One byte = Eight bits

Thirty-two bits (4 x 8), or 4 bytes

Source: https://upload.wikimedia.org/wikipedia/commons/7/74/Ipv4_address.svg



An IPv6 address (in hexadecimal)

2001:0DB8:AC10:FE01:0000:0000:0000:0000

2001:0DB8:AC10:FE01::

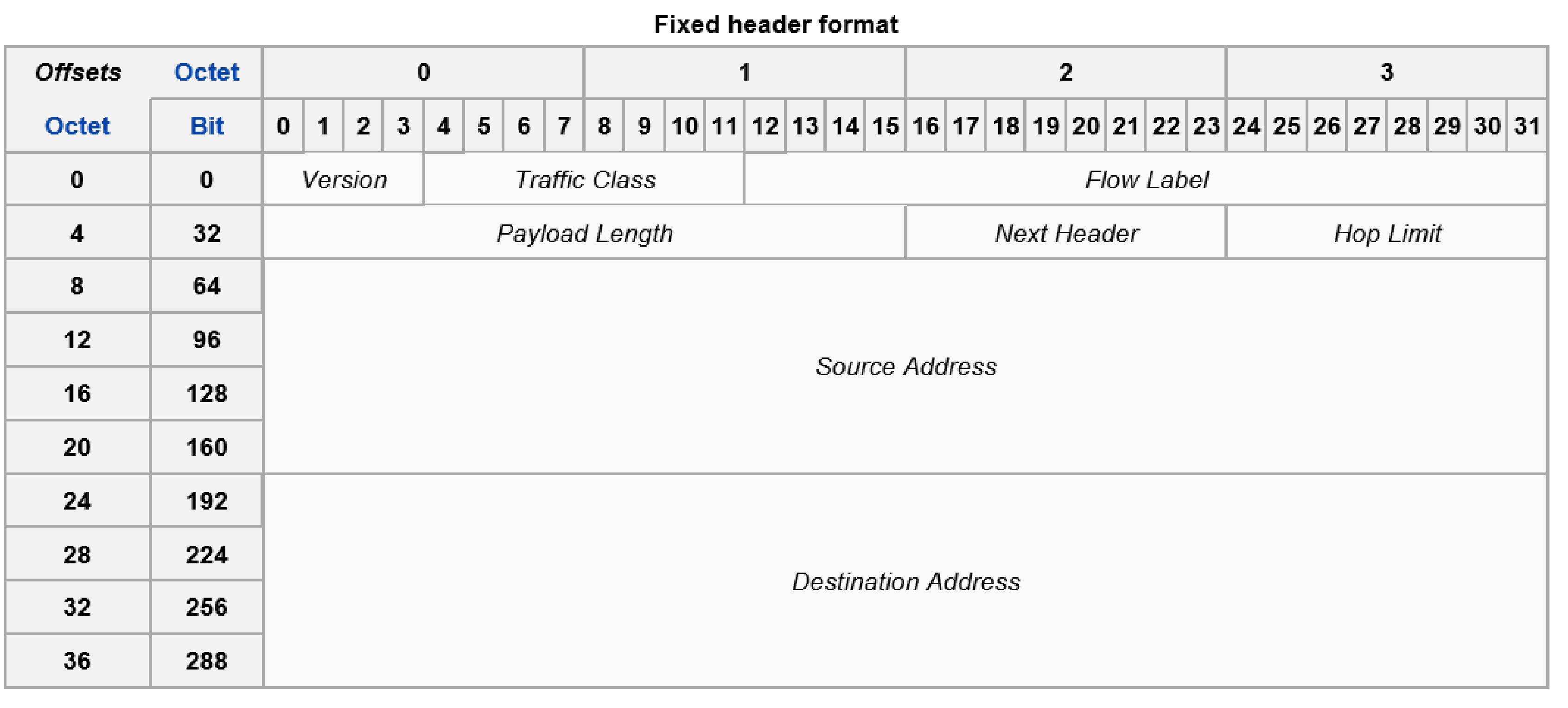
Zeroes can be omitted

0010000000000001:0000110110111000:1010110000010000:1111111000000001:

0000000000000000:0000000000000000:0000000000000000:0000000000000000

Source: https://en.wikipedia.org/wiki/File:Ipv6_address_leading_zeros.svg

IPv6



Fixed header format

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				Traffic Class								Flow Label																			
4	32	Payload Length																Next Header								Hop Limit							
8	64	Source Address																															
12	96																																
16	128																																
20	160																																
24	192	Destination Address																															
28	224																																
32	256																																
36	288																																

SSL

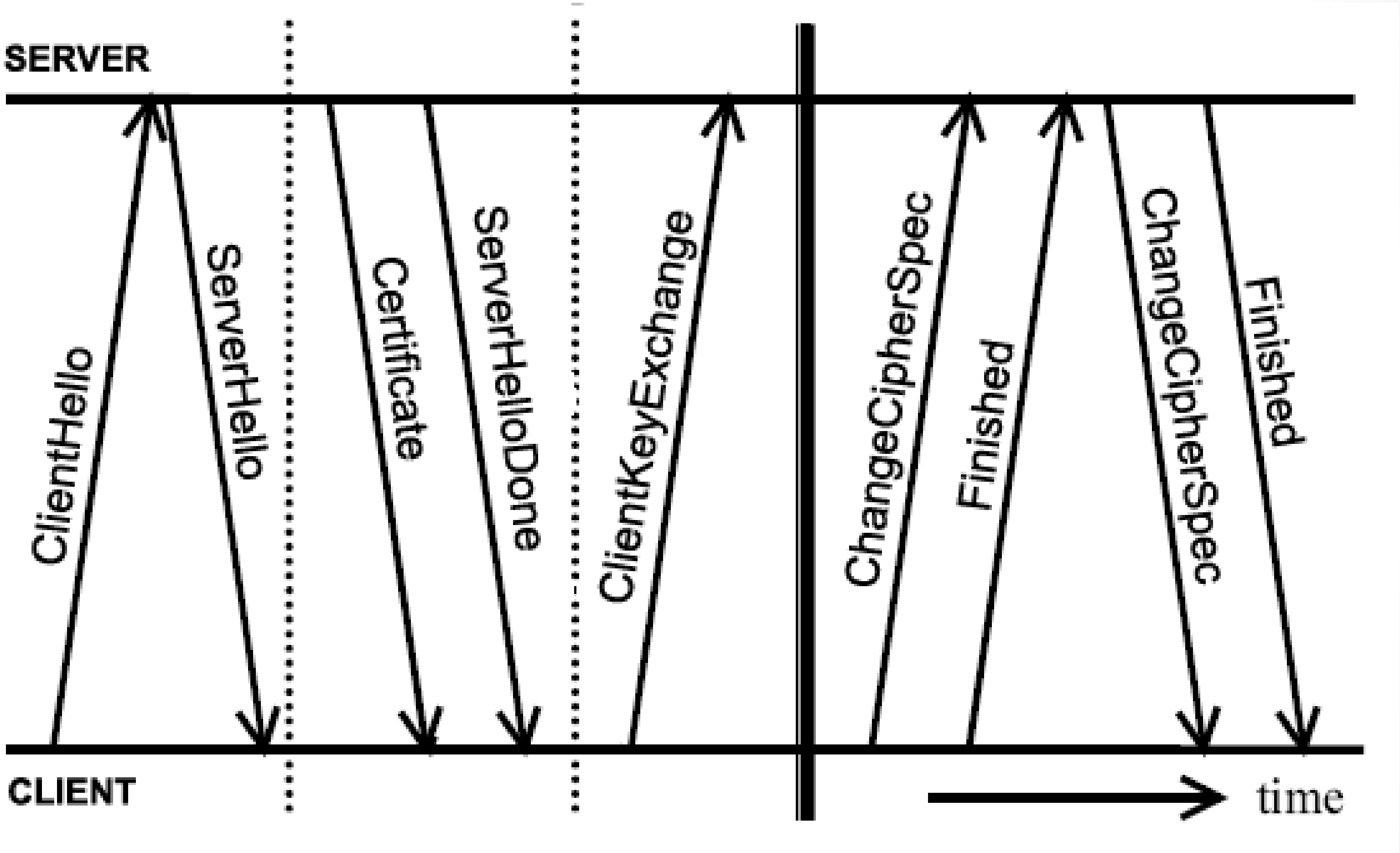
- Secure Socket Layer (SSL) is a security socket connection that provides a security layer at the transport level between the TCP/IP transport and sockets.
- The objective is to securely transmit from one site to another without involving the applications that invoke it
- The SSL protocol provides a certificate-based server authentication, private client-server communications using Rivest-Shamir-Adleman (RSA) encryption and message integrity checks.
- The SSL client generates a secret key for one session that is encrypted using the server's public key.
  - The session key is forwarded to the server and used for communication between the client and the server.

http://tools.ietf.org/html/rfc6101

SSL

- Basic properties
  - The connection is private.
    - Encryption is used after an initial handshake to define a secret key.
    - Symmetric cryptography is used for data encryption.
      - DES, 3DES, RC4
  - The peer's identity can be authenticated using asymmetric, or public key, cryptography.
    - RSA, DSS
  - The connection is reliable.
    - Message transport includes a message integrity check using a keyed Message Authentication Code (MAC) [RFC2104].
    - Secure hash functions (e.g., SHA, MD5) are used for MAC computations.

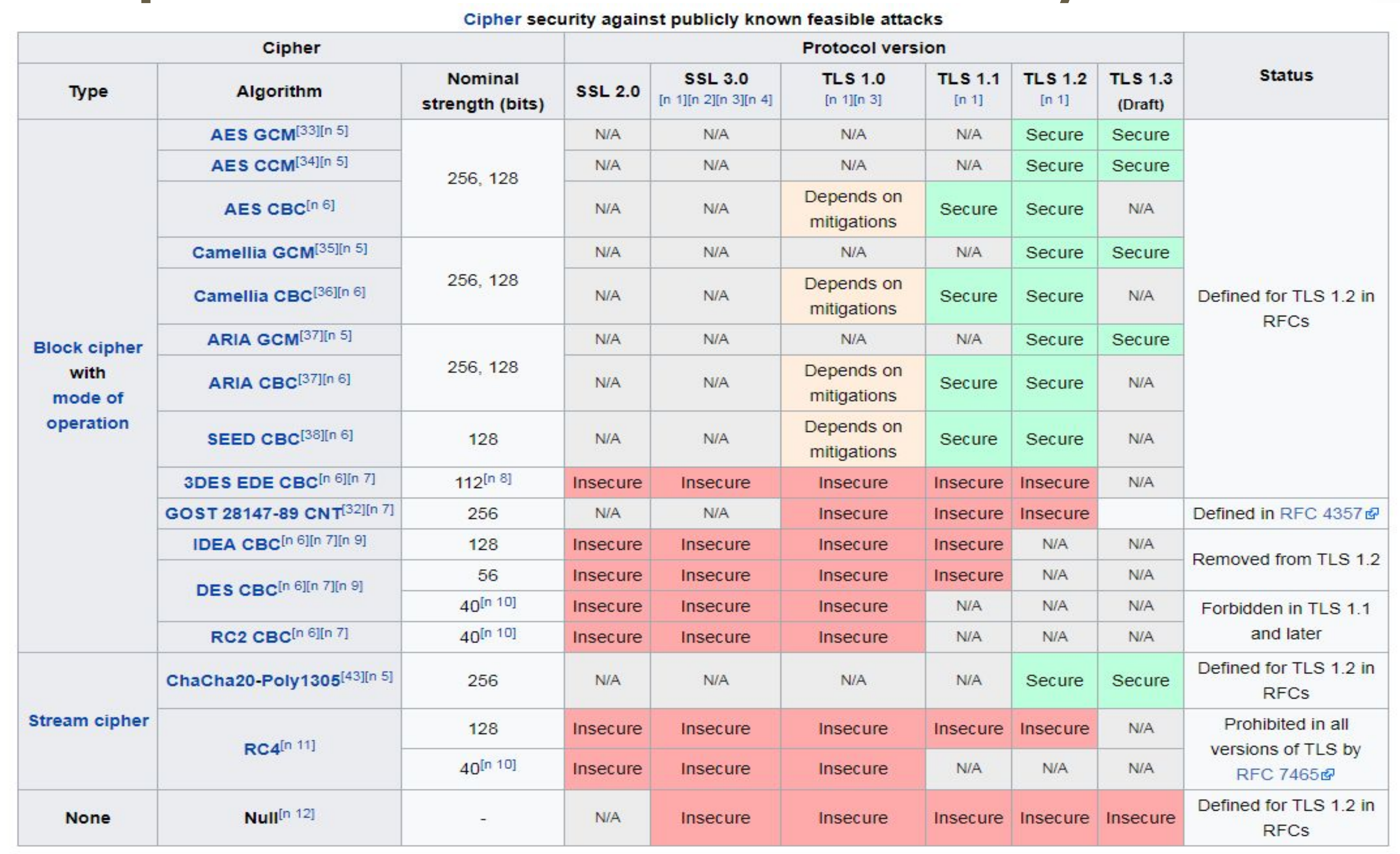
SSL


SERVER
CLIENT
ClientHello
ServerHello
Certificate
ServerHelloDone
ClientKeyExchange
ChangeCipherSpec
Finished
ChangeCipherSpec
Finished
time

Transport Layer Security (TLS)

- "TLS versions 1.0, 1.1, and 1.2, and SSL 3.0 are very similar" [http://tools.ietf.org/html/rfc5246, The Transport Layer Security (TLS) Protocol, Version 1.2, 2008].
  - There are many minor differences between these protocols, but browsers and servers are often configured to "rollback" to an earlier protocol in this family – if their communication partner requests this.
  - Attackers may exploit the differences and the rollbacks, see https://www.ietf.org/proceedings/84/slides/slides-84-tls-4.pdf
- Most experts advise against using the older protocols.
  - Qualys deprecates any browser that accepts SSL2.0, see https://www.ssllabs.com/ssltest/viewMyClient.html and https://www.ssllabs.com/projects/rating-guide/
- "SSL/TLS is a deceptively simple technology.
  - "It is easy to deploy, and it just works . . . except that it does not, really.
  - The first part is true—SSL is easy to deploy—but it turns out that it is not easy to deploy correctly." [https://www.ssllabs.com/projects/best-practices/]

Wikipedia's Current Advice on Cipher Selection in SSL/TLS

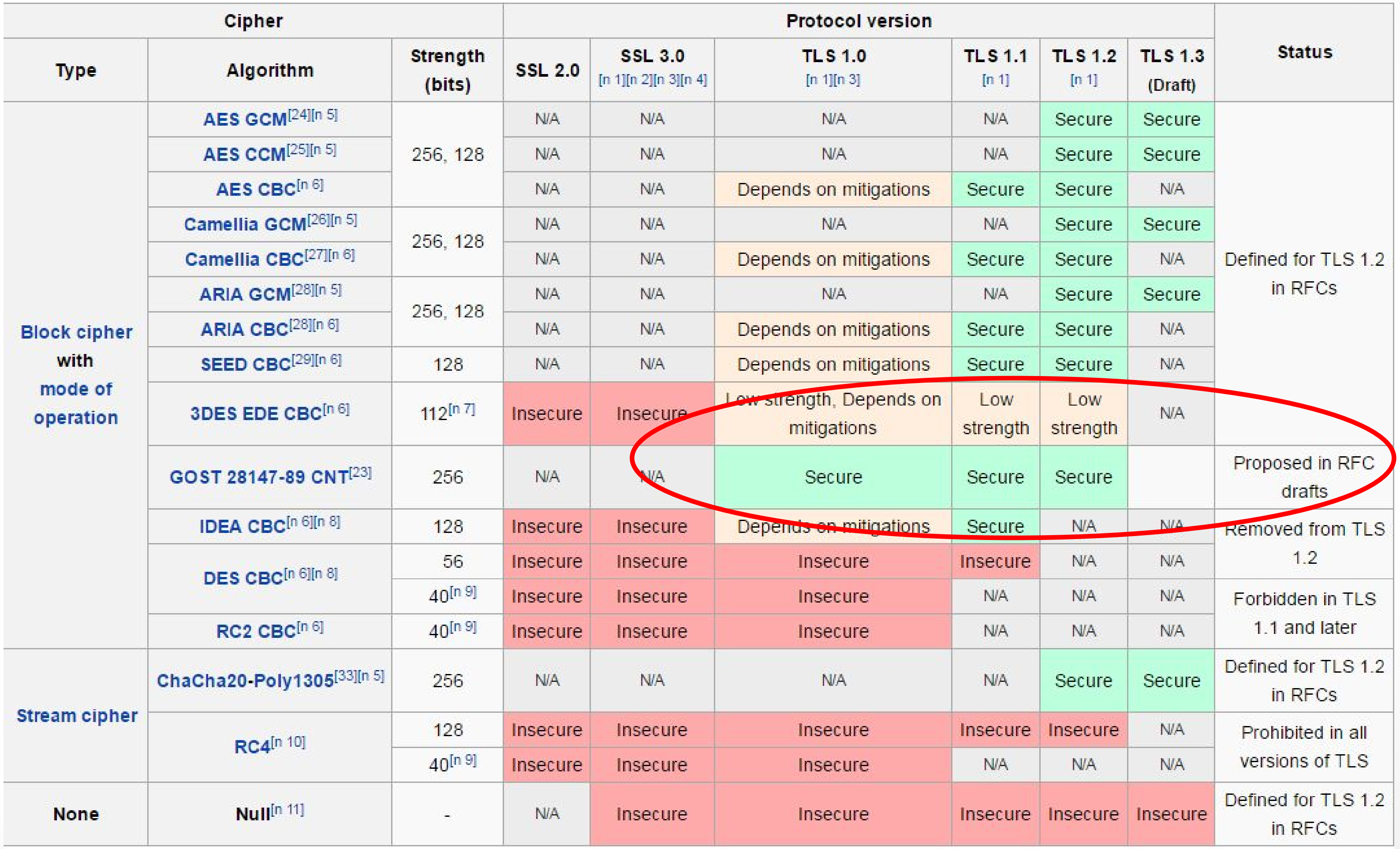


Cipher security against publicly known feasible attacks

Cipher			Protocol version						Status
Type	Algorithm	Nominal strength (bits)	SSL 2.0	SSL 3.0 [n 1][n 2][n 3][n 4]	TLS 1.0 [n 1][n 3]	TLS 1.1 [n 1]	TLS 1.2 [n 1]	TLS 1.3 (Draft)	
Block cipher with mode of operation	AES GCM[33][n 5]	256, 128	N/A	N/A	N/A	N/A	Secure	Secure	Defined for TLS 1.2 in RFCs
	AES CCM[34][n 5]		N/A	N/A	N/A	N/A	Secure	Secure	
	AES CBC[n 6]		N/A	N/A	Depends on mitigations	Secure	Secure	N/A	
	Camellia GCM[35][n 5]	256, 128	N/A	N/A	N/A	N/A	Secure	Secure	
	Camellia CBC[36][n 6]		N/A	N/A	Depends on mitigations	Secure	Secure	N/A	
	ARIA GCM[37][n 5]	256, 128	N/A	N/A	N/A	N/A	Secure	Secure	
	ARIA CBC[37][n 6]		N/A	N/A	Depends on mitigations	Secure	Secure	N/A	
	SEED CBC[38][n 6]	128	N/A	N/A	Depends on mitigations	Secure	Secure	N/A	
	3DES EDE CBC[n 6][n 7]	112[n 8]	Insecure	Insecure	Insecure	Insecure	Insecure	N/A	
	GOST 28147-89 CNT[32][n 7]	256	N/A	N/A	Insecure	Insecure	Insecure		Defined in RFC 4357
	IDEA CBC[n 6][n 7][n 9]	128	Insecure	Insecure	Insecure	Insecure	N/A	N/A	Removed from TLS 1.2
	DES CBC[n 6][n 7][n 9]	56	Insecure	Insecure	Insecure	Insecure	N/A	N/A	
		40[n 10]	Insecure	Insecure	Insecure	N/A	N/A	N/A	Forbidden in TLS 1.1 and later
	RC2 CBC[n 6][n 7]	40[n 10]	Insecure	Insecure	Insecure	N/A	N/A	N/A	
Stream cipher	ChaCha20-Poly1305[43][n 5]	256	N/A	N/A	N/A	N/A	Secure	Secure	Defined for TLS 1.2 in RFCs
	RC4[n 11]	128	Insecure	Insecure	Insecure	Insecure	Insecure	N/A	Prohibited in all versions of TLS by RFC 7465
		40[n 10]	Insecure	Insecure	Insecure	N/A	N/A	N/A	
None	Null[n 12]	-	N/A	Insecure	Insecure	Insecure	Insecure	Insecure	Defined for TLS 1.2 in RFCs

http://en.wikipedia.org/wiki/Transport_Layer_Security, 14 August 2017

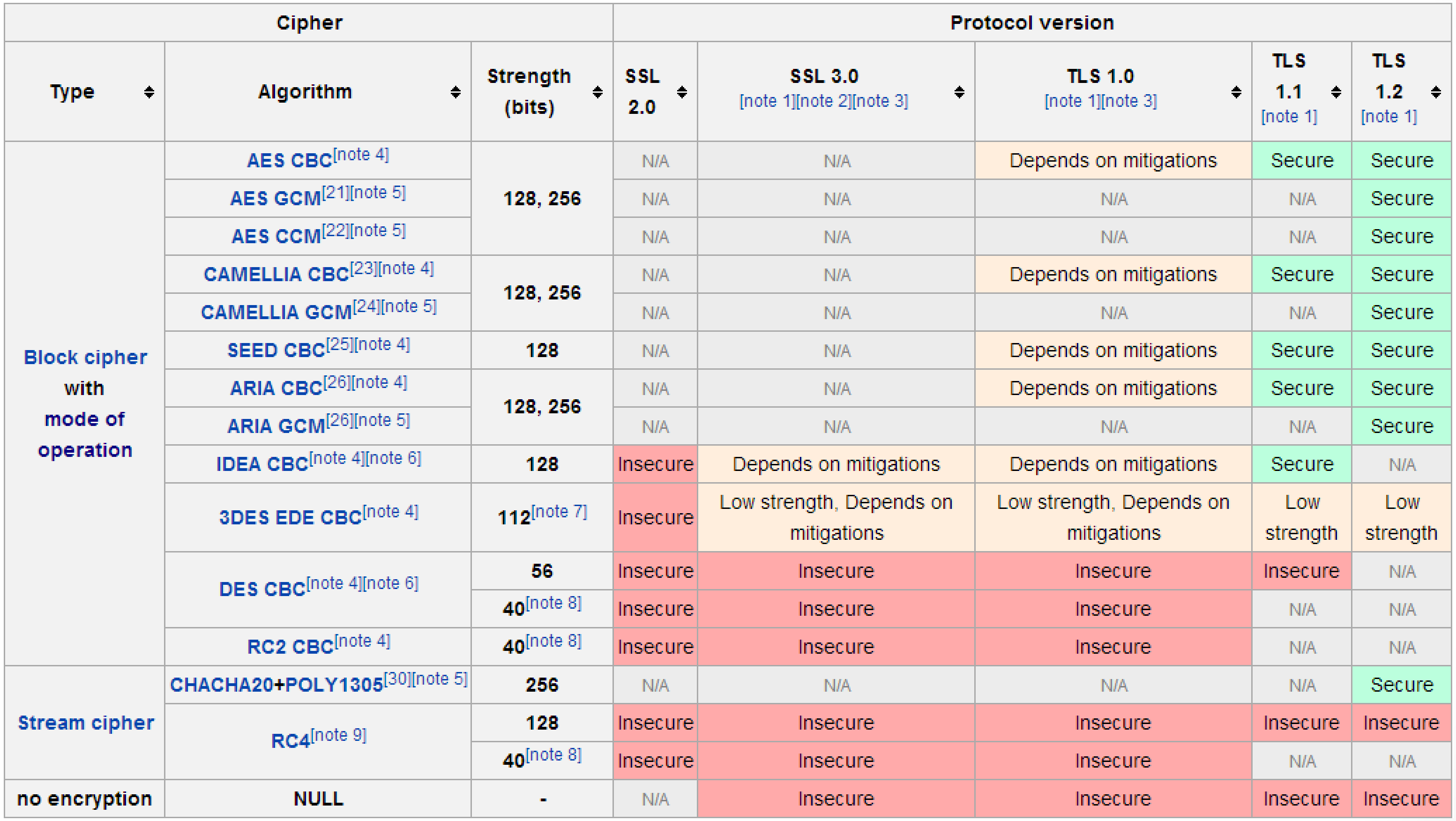
Wikipedia’s 2016 Advice on Cipher Selection in SSL/TLS



Cipher			Protocol version						Status
Type	Algorithm	Strength (bits)	SSL 2.0	SSL 3.0 [n 1][n 2][n 3][n 4]	TLS 1.0 [n 1][n 3]	TLS 1.1 [n 1]	TLS 1.2 [n 1]	TLS 1.3 (Draft)	
Block cipher with mode of operation	AES GCM[24][n 5]	256, 128	N/A	N/A	N/A	N/A	Secure	Secure	Defined for TLS 1.2 in RFCs
	AES CCM[25][n 5]		N/A	N/A	N/A	N/A	Secure	Secure	
	AES CBC[n 6]		N/A	N/A	Depends on mitigations	Secure	Secure	N/A	
	Camellia GCM[26][n 5]	256, 128	N/A	N/A	N/A	N/A	Secure	Secure	
	Camellia CBC[27][n 6]		N/A	N/A	Depends on mitigations	Secure	Secure	N/A	
	ARIA GCM[28][n 5]	256, 128	N/A	N/A	N/A	N/A	Secure	Secure	
	ARIA CBC[28][n 6]		N/A	N/A	Depends on mitigations	Secure	Secure	N/A	
	SEED CBC[29][n 6]	128	N/A	N/A	Depends on mitigations	Secure	Secure	N/A	
	3DES EDE CBC[n 6]	112[n 7]	Insecure	Insecure	Low strength, Depends on mitigations	Low strength	Low strength	N/A	
	GOST 28147-89 CNT[23]	256	N/A	N/A	Secure	Secure	Secure		Proposed in RFC drafts
	IDEA CBC[n 6][n 8]	128	Insecure	Insecure	Depends on mitigations	Secure	N/A	N/A	Removed from TLS 1.2
	DES CBC[n 6][n 8]	56	Insecure	Insecure	Insecure	Insecure	N/A	N/A	
		40[n 9]	Insecure	Insecure	Insecure	N/A	N/A	N/A	Forbidden in TLS 1.1 and later
	RC2 CBC[n 6]	40[n 9]	Insecure	Insecure	Insecure	N/A	N/A	N/A	
Stream cipher	ChaCha20-Poly1305[33][n 5]	256	N/A	N/A	N/A	N/A	Secure	Secure	Defined for TLS 1.2 in RFCs
	RC4[n 10]	128	Insecure	Insecure	Insecure	Insecure	Insecure	N/A	Prohibited in all versions of TLS
		40[n 9]	Insecure	Insecure	Insecure	N/A	N/A	N/A	
None	Null[n 11]	-	N/A	Insecure	Insecure	Insecure	Insecure	Insecure	Defined for TLS 1.2 in RFCs

http://en.wikipedia.org/wiki/Transport_Layer_Security, 27 July 2016

Wikipedia's 2014 Advice on Cipher Selection in SSL/TLS

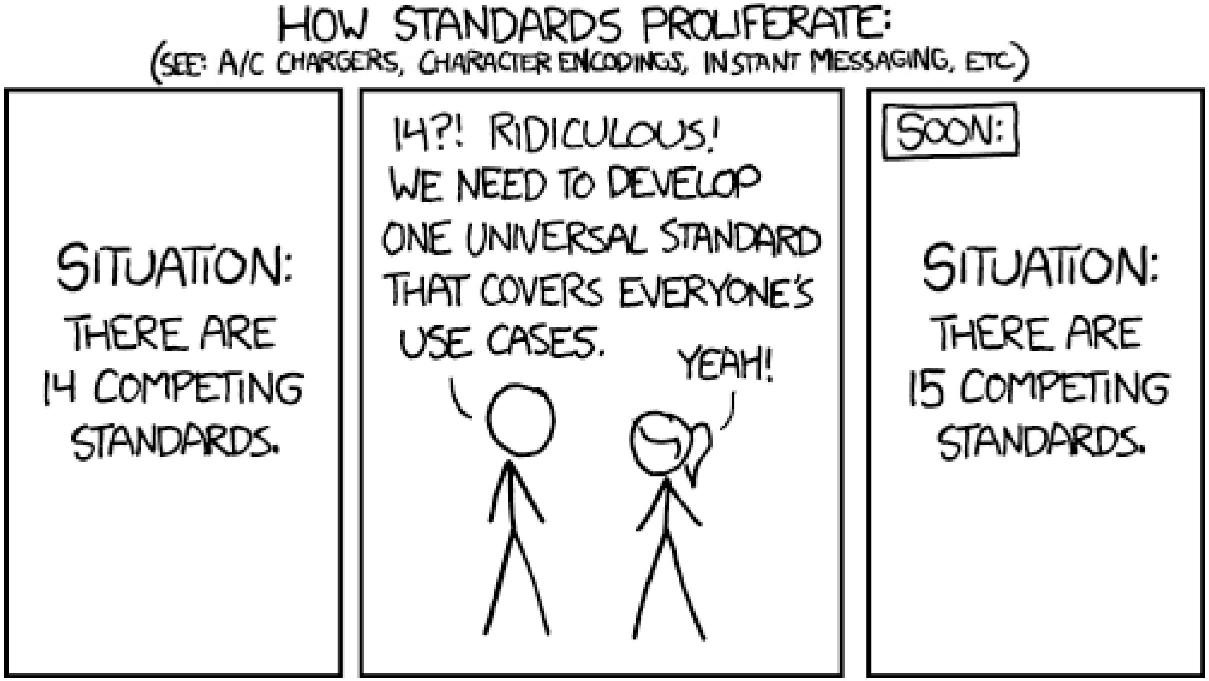


Cipher			Protocol version				
Type	Algorithm	Strength (bits)	SSL 2.0	SSL 3.0 [note 1][note 2][note 3]	TLS 1.0 [note 1][note 3]	TLS 1.1 [note 1]	TLS 1.2 [note 1]
Block cipher with mode of operation	AES CBC[note 4]	128, 256	N/A	N/A	Depends on mitigations	Secure	Secure
	AES GCM[21][note 5]		N/A	N/A	N/A	N/A	Secure
	AES CCM[22][note 5]		N/A	N/A	N/A	N/A	Secure
	CAMELLIA CBC[23][note 4]	128, 256	N/A	N/A	Depends on mitigations	Secure	Secure
	CAMELLIA GCM[24][note 5]		N/A	N/A	N/A	N/A	Secure
	SEED CBC[25][note 4]	128	N/A	N/A	Depends on mitigations	Secure	Secure
	ARIA CBC[26][note 4]	128, 256	N/A	N/A	Depends on mitigations	Secure	Secure
	ARIA GCM[26][note 5]		N/A	N/A	N/A	N/A	Secure
	IDEA CBC[note 4][note 6]	128	Insecure	Depends on mitigations	Depends on mitigations	Secure	N/A
	3DES EDE CBC[note 4]	112[note 7]	Insecure	Low strength, Depends on mitigations	Low strength, Depends on mitigations	Low strength	Low strength
	DES CBC[note 4][note 6]	56	Insecure	Insecure	Insecure	Insecure	N/A
		40[note 8]	Insecure	Insecure	Insecure	N/A	N/A
	RC2 CBC[note 4]	40[note 8]	Insecure	Insecure	Insecure	N/A	N/A
Stream cipher	CHACHA20+POLY1305[30][note 5]	256	N/A	N/A	N/A	N/A	Secure
	RC4[note 9]	128	Insecure	Insecure	Insecure	Insecure	Insecure
		40[note 8]	Insecure	Insecure	Insecure	N/A	N/A
no encryption	NULL	-	N/A	Insecure	Insecure	Insecure	Insecure

http://en.wikipedia.org/wiki/Transport_Layer_Security, 1 August 2014

A Lighthearted View

- Question at https://www.schneier.com/blog/archives/2013/02/really_clever_t.html:
  - “It's probably fair to say that TLS has accrued too many options and versions to remain secure overall.
  - “Time to throw it out and build a new protocol that avoids all the problems identified with TLS over the years.
  - “Who'll go first?”
- Answer: … Time for obligatory xkcd: http://xkcd.com/927/


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

IPSec

- Short for IP Security, a set of protocols developed by the IETF to support the secure exchange of packets at the IP layer.
  - IPsec has been deployed widely to implement Virtual Private Networks (VPNs).
- For IPsec to work, the sending and receiving devices must share a public key.
  - Internet Security Association and Key Management Protocol/Oakley (ISAKMP/Oakley) protocol.
  - Allows the receiver to obtain a public key and authenticate the sender using digital certificates.

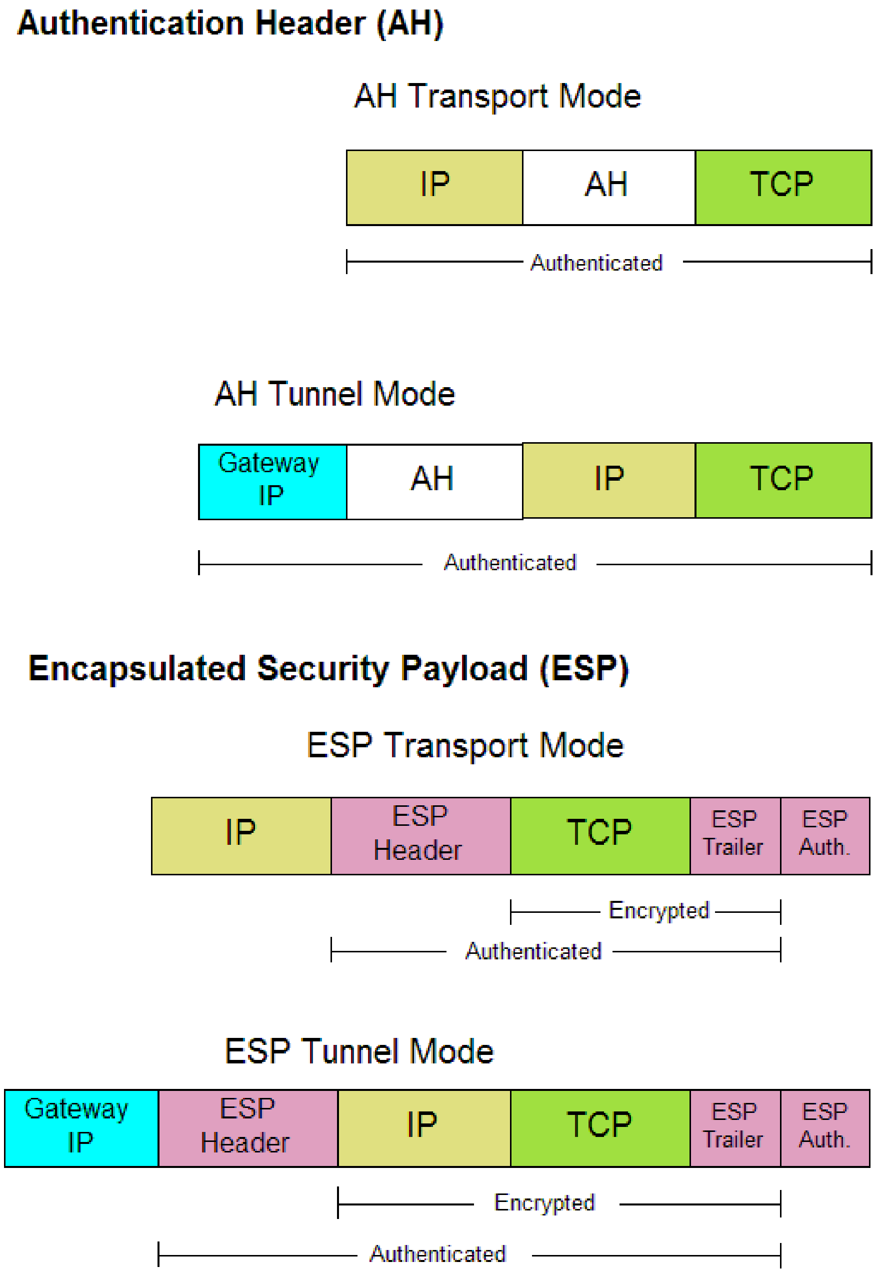
http://tools.ietf.org/html/rfc4301

Category: Data Confidentiality

IPSec

- IPSec may be used to protect one or more paths between two of any combination of hosts and/or security gateways (routers, firewalls, etc).
  - This is facilitated through the use of its Authentication Header (AH), and its Encapsulating Security Payload (ESP), both of which are algorithm independent.
  - The AH is used to authenticate the origin of the packets and the ESP encapsulating the content within the packets
- IPsec supports two encryption modes
  - Transport mode encrypts only the data portion (payload) of each packet, but leaves the header untouched.
  - Tunnel mode encrypts both the header and the payload.
  - On the receiving side, an IPSec-compliant device decrypts each packet.

IPSec



Authentication Header (AH)

AH Transport Mode

IP | AH | TCP

Authenticated

AH Tunnel Mode

Gateway IP | AH | IP | TCP

Authenticated

Encapsulated Security Payload (ESP)

ESP Transport Mode

IP | ESP Header | TCP | ESP Trailer | ESP Auth.

Encrypted

Authenticated

ESP Tunnel Mode

Gateway IP | ESP Header | IP | TCP | ESP Trailer | ESP Auth.

Encrypted

Authenticated

IPSec

- IKE-Related Output (VeriSign CA enrollment)

```
dt1-45a#show crypto key mypubkey rsa
% Key pair was generated at: 11:31:59 PDT Apr 9 1998
Key name: dt1-45a.cisco.com
 Usage: Signature Key
 Key Data:
  305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00C11854 39A9C75C
  4E34C987 B4D7F36C A058D697 13172767 192166E1 661483DD 0FDB907B F9C10B7A
  CB5A034F A41DF385 23BEB6A7 C14344BE E6915A12 1C86374F 83020301 0001
% Key pair was generated at: 11:32:02 PDT Apr 9 1998
Key name: dt1-45a.cisco.com
 Usage: Encryption Key
 Key Data:
  305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00DCF5AC 360DD5A6
C69704CF 47B2362D 65123BD4 424B6FF6 AD10C33E 89983D08 16F1EA58 3700BCF9
  1EF17E71 5931A9FC 18D60D9A E0852DDD 3F25369C F09DFB75 05020301 0001
```

http://www.cisco.com/c/en/us/support/docs/security-vpn/ipsec-negotiation-ike-protocols/16439-IPSECpart8.html

Final Thoughts

- There are many many many more protocols and standards than presented here
  - You can spend an entire lifetime studying this stuff
    - Many folks have done so…
- Lots of discussion… which is the point.
  - Important security protocols are either implementations of standards, or are de-facto standards.
    - Standards can be vague, biased or ineffective; with multiple versions
- Don’t take anything as the absolute unchanging truth
  - Read the archival sources e.g. http://www.ietf.org/rfc.html