**4. Find and document information for the following security hash algorithms MD5 and SHA.**

***SHA***In cryptography, SHA-1 is a cryptographic hash function designed by the United States National Security Agency and published by the United States NIST as a U.S. Federal Information Processing Standard.

SHA-1 produces a 160-bit (20-byte) hash value. A SHA-1 hash value is typically expressed as a hexadecimal number, 40 digits long.

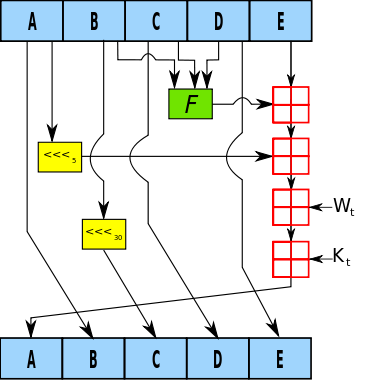
SHA stands for "secure hash algorithm". The four SHA algorithms are structured differently and are distinguished as SHA-0, SHA-1, SHA-2, and SHA-3. SHA-1 is very similar to SHA-0, but corrects an error in the original SHA hash specification that led to significant weaknesses. The SHA-0 algorithm was not adopted by many applications. SHA-2 on the other hand significantly differs from the SHA-1 hash function.

SHA-1 is the most widely used of the existing SHA hash functions, and is employed in several widely used applications and protocols.

In 2005, cryptanalysts found attacks on SHA-1 suggesting that the algorithm might not be secure enough for ongoing use. NIST required many applications in federal agencies to move to SHA-2 after 2010 because of the weakness. Although no successful attacks have yet been reported on SHA-2, they are algorithmically similar to SHA-1. In 2012, following a long-running competition, NIST selected an additional algorithm, Keccak, for standardization as SHA-3. In 2013 Microsoft announced their deprecation policy on SHA-1 according to which Windows will stop accepting SHA-1 certificates in SSL by 2017.

***The SHA-1 hash function***SHA-1 produces a message digest based on principles similar to those used by Ronald L. Rivest of MIT in the design of the MD4 and MD5 message digest algorithms, but has a more conservative design.

The original specification of the algorithm was published in 1993 as the Secure Hash Standard, FIPS PUB 180, by U.S. government standards agency NIST (National Institute of Standards and Technology). This version is now often referred to as SHA-0. It was withdrawn by the NSA shortly after publication and was superseded by the revised version, published in 1995 in FIPS PUB 180-1 and commonly referred to as SHA-1. SHA-1 differs from SHA-0 only by a single bitwise rotation in the message schedule of its compression function; this was done, according to the NSA, to correct a flaw in the original algorithm which reduced its cryptographic security. However, the NSA did not provide any further explanation or identify the flaw that was corrected. Weaknesses have subsequently been reported in both SHA-0 and SHA-1. SHA-1 appears to provide greater resistance to attacks[citation needed], supporting the NSA’s assertion that the change increased the security.

  
**One iteration within the SHA-1 compression function:**

**A, B, C, D and E are 32-bit words of the state;**

**F is a nonlinear function that varies;**

**n denotes a left bit rotation by n places;**

**n varies for each operation;**

**Wt is the expanded message word of round t;**

**Kt is the round constant of round t;**

**denotes addition modulo 232.**

***MD5***

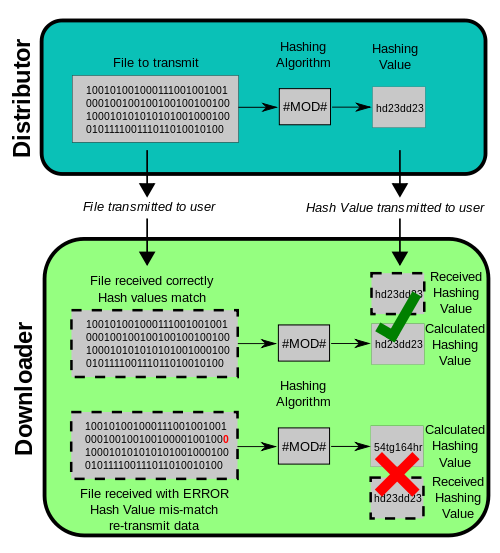
The MD5 message-digest algorithm is a widely used cryptographic hash function producing a 128-bit (16-byte) hash value, typically expressed in text format as a 32 digit hexadecimal number. MD5 has been utilized in a wide variety of cryptographic applications, and is also commonly used to verify data integrity.

MD5 was designed by Ron Rivest in 1991 to replace an earlier hash function, MD4.

In 1996 a flaw was found in the design of MD5. While it was not deemed a fatal weakness at the time, cryptographers began recommending the use of other algorithms, such as SHA-1—which has since been found to be vulnerable as well. In 2004 it was shown that MD5 is not collision resistant. As such, MD5 is not suitable for applications like SSL certificates or digital signatures that rely on this property for digital security. Also in 2004 more serious flaws were discovered in MD5, making further use of the algorithm for security purposes questionable; specifically, a group of researchers described how to create a pair of files that share the same MD5 checksum. Further advances were made in breaking MD5 in 2005, 2006, and 2007. In December 2008, a group of researchers used this technique to fake SSL certificate validity, and CMU Software Engineering Institute now says that MD5 "should be considered cryptographically broken and unsuitable for further use", and most U.S. government applications now require the SHA-2 family of hash functions. In 2012, the Flame malware exploited the weaknesses in MD5 to fake a Microsoft digital signature.

***Security***The security of the MD5 hash function is severely compromised. A collision attack exists that can find collisions within seconds on a computer with a 2.6 GHz Pentium 4 processor (complexity of 224.1). Further, there is also a chosen-prefix collision attack that can produce a collision for two inputs with specified prefixes within hours, using off-the-shelf computing hardware (complexity 239). The ability to find collisions has been greatly aided by the use of off-the-shelf GPUs. On an NVIDIA GeForce 8400GS graphics processor, 16–18 million hashes per second can be computed. An NVIDIA GeForce 8800 Ultra can calculate more than 200 million hashes per second.

***Applications***MD5 digests have been widely used in the software world to provide some assurance that a transferred file has arrived intact. For example, file servers often provide a pre-computed MD5 (known as Md5sum) checksum for the files, so that a user can compare the checksum of the downloaded file to it. Most unix-based operating systems include MD5 sum utilities in their distribution packages; Windows users may install a Microsoft utility, or use third-party applications. Android ROMs also utilize this type of checksum.



However, now that it is easy to generate MD5 collisions, it is possible for the person who created the file to create a second file with the same checksum, so this technique cannot protect against some forms of malicious tampering. Also, in some cases, the checksum cannot be trusted (for example, if it was obtained over the same channel as the downloaded file), in which case MD5 can only provide error-checking functionality: it will recognize a corrupt or incomplete download, which becomes more likely when downloading larger files.

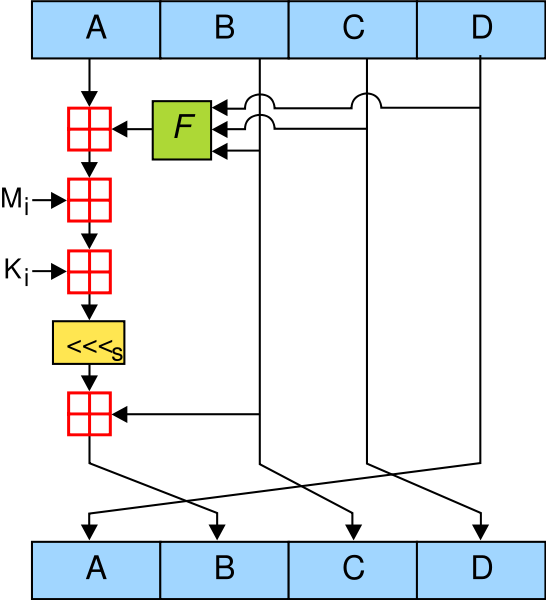
MD5 can be used to store a one-way hash of a password, often with key stretching.Along with other hash functions, it is also used in the field of electronic discovery, in order to provide a unique identifier for each document that is exchanged during the legal discovery process. This method can be used to replace the Bates stamp numbering system that has been used for decades during the exchange of paper documents.

***Algorithm***MD5 processes a variable-length message into a fixed-length output of 128 bits. The input message is broken up into chunks of 512-bit blocks (sixteen 32-bit words); the message is padded so that its length is divisible by 512. The padding works as follows: first a single bit, 1, is appended to the end of the message. This is followed by as many zeros as are required to bring the length of the message up to 64 bits fewer than a multiple of 512. The remaining bits are filled up with 64 bits representing the length of the original message, modulo 264.

The main MD5 algorithm operates on a 128-bit state, divided into four 32-bit words, denoted A, B, C and D. These are initialized to certain fixed constants. The main algorithm then uses each 512-bit message block in turn to modify the state. The processing of a message block consists of four similar stages, termed rounds; each round is composed of 16 similar operations based on a non-linear function F, modular addition, and left rotation. Figure 1 illustrates one operation within a round. There are four possible functions F; a different one is used in each round:


 denote the XOR, AND, OR and NOT operations respectively.

  
**One MD5 operation. MD5 consists of 64 of these operations, grouped in four rounds of 16 operations. F is a nonlinear function; one function is used in each round. Mi denotes a 32-bit block of the message input, and Ki denotes a 32-bit constant, different for each operation. s denotes a left bit rotation by s places; s varies for each operation. denotes addition modulo 232.**