# Cryptography General Job Interview Questions And Answers

**Interview Questions Answers**

**https://interviewquestionsanswers.org/**

# About Interview Questions Answers

**Interview Questions Answers . ORG** is an interview preparation guide of thousands of Job Interview Questions And Answers, Job Interviews are always stressful even for job seekers who have gone on countless interviews. The best way to reduce the stress is to be prepared for your job interview. Take the time to review the standard interview questions you will most likely be asked. These interview questions and answers on Cryptography General will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts.

If you find any **question or answer** is incorrect or incomplete then you can **submit your question or answer** directly with out any registration or login at our website. You just need to visit Cryptography General Interview Questions And Answers to add your answer click on the *Submit Your Answer* links on the website; with each question to post your answer, if you want to ask any question then you will have a link *Submit Your Question*; that's will add your question in Cryptography General category. To ensure quality, each submission is checked by our team, before it becomes live. This Cryptography General Interview preparation PDF was generated at **Wednesday 29th November, 2023**

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.
www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter for latest Jobs and interview preparation guides.
https://twitter.com/InterviewQA

If you need any further assistance or have queries regarding this document or its material or any of other inquiry, please do not hesitate to contact us.

Best Of Luck.

**Interview Questions Answers.ORG Team**
**https://InterviewQuestionsAnswers.ORG/**
**Support@InterviewQuestionsAnswers.ORG**

# Cryptography General Interview Questions And Answers Guide.

**Question - 1:**

What is the Quantum Cryptography?

**Ans:**

Quantum cryptography is a method for secure key exchange over an insecure channel based on the nature of photons. Photons have a polarization, which can be measured in any basis, where a basis consists of two directions orthogonal to each other. If a photon's polarization is read in the same basis twice, the polarization will be read correctly and will remain unchanged. If it is read in two different bases, a random answer will be obtained in the second basis, and the polarization in the initial basis will be changed randomly.
The following protocol can be used by Alice and Bob to exchange secret keys.
Alice sends Bob a stream of photons, each with a random polarization, in a random basis. She records the polarizations.
Bob measures each photon in a randomly chosen basis and records the results.
Bob announces, over an authenticated but not necessarily private channel (e.g., by telephone), which basis he used for each photon.
Alice tells him which choices of bases are correct.

View All Answers

**Question - 2:**

What is DNA Computing?

**Ans:**

DNA computing, also known as molecular computing, is a new approach to massively parallel computation based on ground-breaking work by Adleman. He used DNA to solve a seven-node Hamiltonian path problem, a special case of an NP-complete problem that attempts to visit every node in a graph exactly once. (This special case is trivial to solve with a conventional computer, or even by hand, but illustrates the potential of DNA computing.)
A DNA computer is basically a collection of specially selected DNA strands whose combinations will result in the solution to some problem. Technology is currently available both to select the initial strands and to filter the final solution. The promise of DNA computing is massive parallelism: with a given setup and enough DNA, one can potentially solve huge problems by parallel search. This can be much faster than a conventional computer, for which massive parallelism would require large amounts of hardware, not simply more DNA.

View All Answers

**Question - 3:**

How do Digital Timestamps Support Digital Signatures?

**Ans:**

Consider two questions that may be asked by a computer user as he or she views a digital document or on-line record. (1) Who is the author of this record - who wrote it, approved it, or consented to it? (2) When was this record created or last modified?
In both cases, the question is one about exactly this record-exactly this sequence of bits - whether it was first stored on this computer or was created somewhere else and then copied and saved here. An answer to the first question tells who & what: who approved exactly what is in this record? An answer to the second question tells when & what: when exactly did the contents of this record first exist?
Both of the above questions have good solutions. A system for answering the first question is called a digital signature scheme. Such a system was first proposed in and there is a wide variety of accepted designs for an implementation of this kind of system.

View All Answers

**Question - 4:**

What are Interactive Proofs and Zero-Knowledge Proofs?

**Ans:**

Informally, an interactive proof is a protocol between two parties in which one party, called the prover, tries to prove a certain fact to the other party, called the verifier. An interactive proof usually takes the form of a challenge-response protocol, in which the prover and the verifier exchange messages and the verifier outputs either "accept" or "reject" at the end of the protocol. Besides their theoretical interests, interactive proofs have found applications in cryptography and computer security such as identification and authentication. In these situations, the fact to be proved is usually related to the prover's identity, e.g., the prover's private key.
The following properties of interactive proofs are quite useful, especially in cryptographic applications:
Completeness: The verifier always accepts the proof if the prover knows the fact and both the prover and the verifier follow the protocol.
Soundness: The verifier always rejects the proof if the prover does not know the fact, as long as the verifier follows the protocol.
Zero knowledge: The verifier learns nothing about the fact being proved (except that it is correct) from the prover that he could not already learn without the

prover. In a zero-knowledge proof, the verifier cannot even later prove the fact to anyone else.
A typical round in a zero-knowledge proof consists of a "commitment" message from the prover, followed by a challenge from the verifier, and then a response to the challenge from the prover. The protocol may be repeated for many rounds. Based on the prover's responses in all the rounds, the verifier decides whether to accept or reject the proof.

View All Answers

**Question - 5:**

What are Visual Secret Sharing Schemes?

**Ans:**

Naor and Shamir developed what they called visual secret sharing schemes, which are an interesting visual variant of the ordinary secret sharing schemes.
Roughly speaking, the problem can be formulated as follows: There is a secret picture to be shared among n participants. The picture is divided into n transparencies (shares) such that if any m transparencies are placed together, the picture becomes visible, but if fewer than m transparencies are placed together, nothing can be seen. Such a scheme is constructed by viewing the secret picture as a set of black and white pixels and handling each pixel separately. See for more details. The schemes are perfectly secure and easily implemented without any cryptographic computation. A further improvement allows each transparency (share) to be an innocent picture (e.g. a picture of a landscape or a picture of a building), thus concealing the fact that secret sharing is taking place.

View All Answers

**Question - 6:**

What is Blakleys Secret Sharing Scheme?

**Ans:**

Blakley's secret sharing scheme is geometric in nature. The secret is a point in an m-dimensional space. n shares are constructed with each share defining a hyperplane in this space. By finding the intersection of any m of these planes, the secret (or point of intersection) can be obtained. This scheme is not perfect, as the person with a share of the secret knows that the secret is a point on his hyperplane. Nevertheless, this scheme can be modified to achieve perfect security.This is based on the scenario where two shares are required to recover the secret. A two-dimensional plane is used as only two shares are required to recover the secret. The secret is a point in the plane. Each share is a line that passes through the point. If any two of the shares are put together, the point of intersection, which is the secret, can be easily derived.

View All Answers

**Question - 7:**

What is Shamirs Secret Sharing Scheme?

**Ans:**

Shamir's secret sharing scheme is an interpolating scheme based on polynomial interpolation. An (m - 1)-degree polynomial over the finite field GF(q)

View All Answers

**Question - 8:**

What are Message Authentication Codes (MACs)?

**Ans:**

A message authentication code (MAC) is an authentication tag (also called a checksum) derived by application of an authentication scheme, together with a secret key, to a message. MACs are computed and verified with the same key so they can only be verified by the intended receiver, unlike digital signatures. MACs can be categorized as (1) unconditionally secure, (2) hash function-based, (3) stream cipher-based, or (4) block cipher-based.
Simmons and Stinson proposed an unconditionally secure MAC that is based on encryption with a one-time pad. The ciphertext of the message authenticates itself, as nobody else has access to the one-time pad. However, there has to be some redundancy in the message. An unconditionally secure MAC can also be obtained by use of a one-time secret key.

View All Answers

**Question - 9:**

What Other Hash Functions Are There?

**Ans:**

For a brief overview here, we note that hash functions are often divided into three classes according to their design:
- those built around block ciphers,
- those which use modular arithmetic, and
- those which have what is termed a "dedicated" design.

By building a hash function around a block cipher, it is intended that by using a well-trusted block cipher such as DES a secure and well-trusted hash function can be obtained. The so-called Davies-Meyer hash function is an example of a hash function built around the use of DES.

View All Answers

**Question - 10:**

What is the Secure Hash Algorithm (SHA and SHA-1)?

**Ans:**

The Secure Hash Algorithm (SHA), the algorithm specified in the Secure Hash Standard (SHS), was developed by NIST and published as a federal information processing standard (FIPS PUB 180). SHA-1 was a revision to SHA that was published in 1994. The revision corrected an unpublished flaw in SHA. Its design is very similar to the MD4 family of hash functions developed by Rivest.

View All Answers

**Question - 11:**

What are pseudo-collisions?

**Ans:**

Pseudo-collisions are collisions for the compression function that lies at the heart of an iterative hash function. While collisions for the compression function of a hash function might be useful in constructing collisions for the hash function itself, this is not normally the case. While pseudo-collisions might be viewed as an unfortunate property of a hash function, a pseudo-collision is not equivalent to a collision, and the hash function can still be secure. MD5 is an example of a hash function for which pseudo-collisions have been discovered and yet is still considered secure.

View All Answers

**Question - 12:**

What is a compression function?

**Ans:**

Damg'rd and Merkle greatly influenced cryptographic hash function design by defining a hash function in terms of what is called a compression function. A compression function takes a fixed length input and returns a shorter, fixed-length output. Then a hash function can be defined by means of repeated applications of the compression function until the entire message has been processed. In this process, a message of arbitrary length is broken into blocks of a certain length which depends on the compression function, and "padded" (for security reasons) so that the size of the message is a multiple of the block size. The blocks are then processed sequentially, taking as input the result of the hash so far and the current message block, with the final output being the hash value for the message.

View All Answers

**Question - 13:**

How does the length of a hash value affect security?

**Ans:**

The essential cryptographic properties of a hash function are that it is both one-way and collision-free. The most basic attack we might mount on a hash function is to choose inputs to the hash function at random until either we find some input that will give us the target output value we are looking for (thereby contradicting the one-way property), or we find two inputs that produce the same output (thereby contradicting the collision-free property).
Suppose the hash function produces an n-bit long output. If we are trying to find some input which will produce some target output value y, then since each output is equally likely we expect to have to try 2n possible input values.

View All Answers

**Question - 14:**

What is a birthday attack?

**Ans:**

A birthday attack is a name used to refer to a class of brute-force attacks. It gets its name from the surprising result that the probability that two or more people in a group of 23 share the same birthday is greater than 1/2; such a result is called a birthday paradox.
If some function, when supplied with a random input, returns one of k equally-likely values, then by repeatedly evaluating the function for different inputs, we expect to obtain the same output after about 1.2k1/2. For the above birthday paradox, replace k with 365.

View All Answers

**Question - 15:**

What is a Hash Function?

**Ans:**

A hash function H is a transformation that takes a variable-size input m and returns a fixed-size string, which is called the hash value h (that is, h = H(m)). Hash functions with just this property have a variety of general computational uses, but when employed in cryptography the hash functions are usually chosen to have some additional properties.
The basic requirements for a cryptographic hash function are:

- the input can be of any length,
- the output has a fixed length,
- H(x) is relatively easy to compute for any given x ,
- H(x) is one-way,
- H(x) is collision-free.

A hash function H is said to be one-way if it is hard to invert, where "hard to invert" means that given a hash value h, it is computationally infeasible to find some input x such that H(x) = h.
If, given a message x, it is computationally infeasible to find a message y not equal to x such that H(x) = H(y) then H is said to be a weakly collision-free hash function.
A strongly collision-free hash function H is one for which it is computationally infeasible to find any two messages x and y such that H(x) = H(y).

View All Answers

**Question - 16:**

What is a One-time Pad?

**Ans:**

A one-time pad, sometimes called the Vernam cipher, uses a string of bits that is generated completely at random. The keystream is the same length as the plaintext message and the random string is combined using bitwise exclusive-or with the plaintext to produce the ciphertext. Since the entire keystream is random, an opponent with infinite computational resources can only guess the plaintext if he sees the ciphertext. Such a cipher is said to offer perfect secrecy and the analysis of the one-time pad is seen as one of the cornerstones of modern cryptography.
While the one-time pad saw use during wartime, over diplomatic channels requiring exceptionally high security, the fact that the secret key (which can be used only once) is as long as the message introduces severe key-management problems. While perfectly secure, the one-time pad is impractical.

View All Answers

**Question - 17:**

What Other Stream Ciphers Are There?

**Ans:**

There are a vast number of alternative stream ciphers that have been proposed in cryptographic literature as well as an equally vast number that appear in

implementations and products world-wide. Many are based on the use of LFSRs since such ciphers tend to be more amenable to analysis and it is easier to assess the security that they offer.

Rueppel suggests that there are essentially four distinct approaches to stream cipher design. The first is termed the information-theoretic approach as exemplified by Shannon's analysis of the one-time pad. The second approach is that of system-theoretic design. In essence, the cryptographer designs the cipher along established guidelines which ensure that the cipher is resistant to all known attacks. While there is, of course, no substantial guarantee that future cryptanalysis will be unsuccessful, it is this design approach that is perhaps the most common in cipher design. The third approach is to attempt to relate the difficulty of breaking the stream cipher (where "breaking" means being able to predict the unseen keystream with a success rate better than can be achieved by guessing) to solving some difficult problem. This complexity-theoretic approach is very appealing, but in practice the ciphers that have been developed tend to be rather slow and impractical. The final approach highlighted by Rueppel is that of designing a randomized cipher. Here the aim is to ensure that the cipher is resistant to any practical amount

View All Answers

**Question - 18:**

What are the Shrinking and Self-Shrinking Generators?

**Ans:**

The shrinking generator was developed by Coppersmith, Krawczyk, and Mansour. It is a stream cipher based on the simple interaction between the outputs from two LFSRs. The bits of one output are used to determine whether the corresponding bits of the second output will be used as part of the overall keystream. The shrinking generator is simple and scaleable, and has good security properties. One drawback of the shrinking generator is that the output rate of the keystream will not be constant unless precautions are taken. A variant of the shrinking generator is the self-shrinking generator, where instead of using one output from one LFSR to "shrink" the output of another (as in the shrinking generator), the output of a single LFSR is used to extract bits from the same output. There are as yet no results on the cryptanalysis of either technique.

View All Answers

**Question - 19:**

What are Shift Register Cascades?

**Ans:**

A shift register cascade is a set of LFSRs (see Question 89) connected together in such a way that the behavior of one particular LFSR depends on the behavior of the previous LFSRs in the cascade. This dependent behavior is usually achieved by using one LFSR to control the clock of the following LFSR. For instance one register might be advanced by one step if the preceding register output is 1 and advanced by two steps otherwise. Many different configurations are possible and certain parameter choices appear to offer very good security .

View All Answers

**Question - 20:**

What is a Linear Feedback Shift Register?

**Ans:**

A Linear Feedback Shift Register (LFSR) is a mechanism for generating a sequence of binary bits. The register consists of a series of cells that are set by an initialization vector that is, most often, the secret key. The behavior of the register is regulated by a clock and at each clocking instant, the contents of the cells of the register are shifted right by one position, and the exclusive-or of a subset of the cell contents is placed in the leftmost cell. One bit of output is usually derived during this update procedure.

View All Answers

**Question - 21:**

What is SEAL?

**Ans:**

The Software-optimized Encryption Algorithm (SEAL) was designed by Rogaway and Coppersmith in 1993 as a fast stream cipher for 32-bit machines. SEAL has a rather involved initialization phase during which a large set of tables is initialized using the Secure Hash Algorithm. However, the use of look-up tables during keystream generation helps to achieve a very fast performance with just five instructions required per byte of output generated.

View All Answers

**Question - 22:**

What is a Stream Cipher?

**Ans:**

A stream cipher is a symmetric encryption algorithm. Stream ciphers can be designed to be exceptionally fast, much faster in fact than any block cipher. While block ciphers operate on large blocks of data, stream ciphers typically operate on smaller units of plaintext, usually bits. The encryption of any particular plaintext with a block cipher will result in the same ciphertext when the same key is used. With a stream cipher, the transformation of these smaller plaintext units will vary, depending on when they are encountered during the encryption process.

A stream cipher generates what is called a keystream and encryption is provided by combining the keystream with the plaintext, usually with the bitwise exclusive-or operation. The generation of the keystream can be independent of the plaintext and ciphertext (yielding what is termed a synchronous stream cipher) or it can depend on the data and its encryption (in which case the stream cipher is said to be self-synchronizing). Most stream cipher designs are for synchronous stream ciphers.

View All Answers

**Question - 23:**

What is Skipjack?

**Ans:**

Skipjack is the encryption algorithm contained in the Clipper chip, and it was designed by the NSA. It uses an 80-bit key to encrypt 64-bit blocks of data. Skipjack can be more secure than DES, since it uses 80-bit keys and scrambles the data for 32 steps, or "rounds"; by contrast, DES uses 56-bit keys and scrambles the data for only 16 rounds.

The details of Skipjack are classified. The decision not to make the details of the algorithm publicly available has been widely criticized. Many people are suspicious that Skipjack is not secure, either due to oversight by its designers, or by the deliberate introduction of a secret trapdoor. By contrast, there have been many attempts to find weaknesses in DES over the years, since its details are public. These numerous attempts (and the fact that they have failed) have made people confident in the

security of DES. Since Skipjack is not public, the same scrutiny cannot be applied towards it, and thus a corresponding level of confidence may not arise.

View All Answers

**Question - 24:**

What is FEAL?

**Ans:**

The Fast Data Encipherment Algorithm (FEAL) was presented by Shimizu and Miyaguchi as an alternative to DES. The original cipher (called FEAL-4) was a four-round cryptosystem with a 64-bit block size and a 64-bit key size and it was designed to give high performance in software. Soon a variety of attacks against FEAL-4 were announced including one attack that required only 20 chosen plaintexts. Several results in the cryptanalysis of FEAL-8 (eight-round version) led the designers to introduce a revised version, FEAL-N, where N denoted the number of rounds. Biham and Shamir developed differential cryptanalytic attacks against FEAL-N for up to 31 rounds. In 1994, Ohta and Aoki presented a linear cryptanalytic attack against FEAL-8 that required 225 known plaintexts, and other improvements followed. In the wake of these numerous attacks, FEAL and its derivatives should be considered insecure.

View All Answers

**Question - 25:**

What is SAFER?

**Ans:**

SAFER (Secure And Fast Encryption Routine) is a non-proprietary block cipher developed by Massey in 1993 for Cylink Corporation. It is a byte-oriented algorithm with a 64-bit block size and, in one version, a 64-bit key size. It has a variable number of rounds (maximum of 10), but a minimum of six rounds is recommended. Unlike most recent block ciphers, SAFER has slightly different encryption and decryption procedures. Only byte-based operations are employed to ensure its utility in smart card-based applications that have limited processing power. When first announced, SAFER was intended to be implemented with a key of length 64 bits and it was accordingly named SAFER K-64. Another version of SAFER was designed that could handle 128-bit keys and was named SAFER K-128.

View All Answers

**Question - 26:**

What is Blowfish?

**Ans:**

Blowfish is a 64-bit block cipher developed by Schneier. It is a Feistel cipher and each round consists of a key-dependent permutation and a key-and-data-dependent substitution. All operations are based on exclusive-ors and additions on 32-bit words. The key has a variable length (with a maximum length of 448 bits) and is used to generate several subkey arrays. This cipher was designed specifically for 32-bit machines and is significantly faster than DES. There was an open competition for the cryptanalysis of Blowfish supported by Dr. Dobb's Journal with a $1000 prize. This contest ended in April 1995 and among the results were the discoveries of existence of certain weak keys , an attack against a three-round version of Blowfish, and a differential attack against certain variants of Blowfish. However, Blowfish can still be considered secure, and Schneier has invited cryptanalysts to continue investigating his cipher.

View All Answers

**Question - 27:**

What is a Group Signature?

**Ans:**

A group signature, introduced by Chaum and van Heijst, allows any member of a group to digitally sign a document in a manner such that a verifier can confirm that it came from the group, but does not know which individual in the group signed the document. The protocol allows for the identity of the signer to be discovered, in case of disputes, by a designated group authority who has some auxiliary information. Unfortunately, each time a member of the group signs a document, a new key pair has to be generated for the signer. The generation of new key pairs causes the length of both the group members' secret keys and the designated authority's auxiliary information to grow. This tends to cause the scheme to become unwieldy when used by a group to sign numerous messages or when used for an extended period of time. Some improvements have been made in the efficiency of this scheme.

View All Answers

**Question - 28:**

What is a Fail-stop Signature Scheme?

**Ans:**

A fail-stop signature scheme is a type of signature devised by van Heyst and Pederson [VP92] to protect against the possibility that an enemy may be able to forge a person's signature. It is a variation of the one-time signature scheme, in which only a single message can be signed and protected by a given key at a time. The scheme is based on the discrete logarithm problem. In particular, if an enemy can forge a signature, then the actual signer can prove that forgery has taken place by demonstrating the solution of a supposedly hard problem. Thus the forger's ability to solve that problem is transferred to the actual signer. (The term "fail-stop" refers to the fact that a signer can detect and stop failures, i.e., forgeries. Note that if the enemy obtains an actual copy of the signer's private key, forgery cannot be detected. What the scheme detects are forgeries based on cryptanalysis.)

View All Answers

**Question - 29:**

What is a Designated Confirmer Signature?

**Ans:**

A designated confirmer signature [Cha94] strikes a balance between self-authenticating digital signatures and zero-knowledge proofs. While the former allows anybody to verify a signature, the latter can only convince one recipient at a time of the authenticity of a given document, and only through interaction with the signer. A designated confirmer signature allows certain designated parties to confirm the authenticity of a document without the need for the signer's input. At the same time, without the aid of either the signer or the designated parties, it is not possible to verify the authenticity of a given document. Chaum developed implementations of designated confirmer signatures with one or more confirmers using RSA digital signatures.

View All Answers

**Question - 30:**

What is a Blind Signature Scheme?

**Ans:**

Blind signature schemes, first introduced by Chaum, allow a person to get a message signed by another party without revealing any information about the message to the other party.
Chaum demonstrated the implementation of this concept using RSA signatures as follows: Suppose Alice has a message m that she wishes to have signed by Bob, and she does not want Bob to learn anything about m. Let (n,e) be Bob's public key and (n,d) be his private key. Alice generates a random value r such that gcd(r, n) = 1 and sends to Bob. The value m' is "blinded" by the random value r, and hence Bob can derive no useful information from it. Bob returns the signed value to Alice. Since s'=rmd mod n, Alice can obtain the true signature s of m by computing. Now Alice's message has a signature she could not have obtained on her own. This signature scheme is secure provided that factoring and root extraction remain difficult. However, regardless of the status of these problems the signature scheme is unconditionally "blind" since r is random. The random r does not allow the signer to learn about the message even if the signer can solve the underlying hard problems.

View All Answers

**Question - 31:**

What are Special Signature Schemes?

**Ans:**

Since the time Diffie and Hellman introduced the concept of digital signatures, many signature schemes have been proposed in cryptographic literature. These schemes can be categorized as either conventional digital signature schemes (e.g., RSA, DSA) or special signature schemes depending on their security features.
In a conventional signature scheme (the original model defined by Diffie and Hellman), we generally assume the following situation:
The signer knows the contents of the message that he has signed.
Anyone who knows the public key of the signer can verify the correctness of the signature at any time without any consent or input from the signer. (Digital signature schemes with this property are called self-authenticating signature schemes.)

View All Answers

**Question - 32:**

Is the Use of DSA Covered by Any Patents?

**Ans:**

David Kravitz, former member of the NSA, holds a patent on DSA. Claus P. Schnorr has asserted that his patent covers certain implementations of DSA.

View All Answers

**Question - 33:**

Is DSA Secure?

**Ans:**

DSA is based on the difficulty of computing discrete logarithm. The algorithm is generally considered secure when the key size is large enough. DSS was originally proposed by NIST with a fixed 512-bit key size. After much criticism that this is not secure enough especially for long-term security, NIST revised DSS to allow key sizes up to 1024 bits.
The particular form of the discrete logarithm problem used in DSA is to compute discrete logarithms in certain subgroups in the finite field GF(p) for some prime p. The problem was first proposed for cryptographic use in 1989. Even though no attacks have been reported on this form of the discrete logarithm problem, further analysis is necessary to fully understand the difficulty of the problem.

View All Answers

**Question - 34:**

What are DSA and DSS?

**Ans:**

The Digital Signature Algorithm (DSA) was published by the National Institute of Standards and Technology (NIST) in the Digital Signature Standard (DSS), which is a part of the U.S. government's Capstone project. DSS was selected by NIST, in cooperation with the NSA, to be the digital authentication standard of the U.S. government. The standard was issued on May 19, 1994.
DSA is based on the discrete logarithm problem and derives from cryptosystems proposed by Schnorr and ElGamal. It is for authentication only. For a detailed description of DSA.
In DSA, signature generation is faster than signature verification, whereas in RSA, signature verification is faster than signature generation (if the public and private exponents, respectively, are chosen for this property, which is the usual case). NIST claims that it is an advantage of DSA that signing is faster, but many people in cryptography think that it is better for verification to be the faster operation.

View All Answers

**Question - 35:**

What is Authenticated Diffie-Hellman Key Agreement?

**Ans:**

The authenticated Diffie-Hellman key agreement protocol, or Station-to-Station (STS) protocol, was developed by Diffie, van Oorschot, and Wiener in 1992 [DVW92] to defeat the middleperson attack on the Diffie-Hellman key agreement protocol. The immunity is achieved by allowing the two parties to authenticate themselves to each other by the use of digital signatures and public-key certificates.

View All Answers

**Question - 36:**

Can RSA be Exported from the United States?

**Ans:**

Export of RSA falls under the same U.S. laws as all other cryptographic products.
RSA used for authentication is more easily exported than when it is used for privacy. In the former case, export is allowed regardless of key (modulus) size, although the exporter must demonstrate that the product cannot be easily converted to use for encryption. In the case of RSA used for privacy (encryption), the U.S.

government generally does not allow export if the key size exceeds 512 bits.

View All Answers

**Question - 37:**

Is RSA Patented?

**Ans:**

RSA is patented under U.S. Patent 4,405,829, issued September 20, 1983 and held by RSA Data Security, Inc. of Redwood City, California; the patent expires 17 years after issue, in 2000. RSA Data Security has a standard, royalty-based licensing policy which can be modified for special circumstances. The U.S. government can use RSA without a license because it was invented at MIT with partial government funding.

View All Answers

**Question - 38:**

Is RSA a De Facto Standard?

**Ans:**

RSA is the most widely used public-key cryptosystem today and has often been called a de facto standard. Regardless of the official standards, the existence of a de facto standard is extremely important for the development of a digital economy. If one public-key system is used everywhere for authentication, then signed digital documents can be exchanged between users in different nations using different software on different platforms; this interoperability is necessary for a true digital economy to develop. Adoption of RSA has grown to the extent that standards are being written to accommodate RSA. When the U.S. financial industry was developing standards for digital signatures, it first developed ANSI X9.30 to support the federal requirement of using the Digital Signature Standard. They then modified X9.30 to X9.31 with the emphasis on RSA digital signatures to support the de facto standard of financial institutions.

View All Answers

**Question - 39:**

Is RSA an Official Standard Today?

**Ans:**

RSA is part of many official standards worldwide. The ISO (International Standards Organization) 9796 standard lists RSA as a compatible cryptographic algorithm, as does the ITU-T X.509 security standard. RSA is part of the Society for Worldwide Interbank Financial Telecommunications (SWIFT) standard, the French financial industry's ETEBAC 5 standard, and the ANSI X9.31 draft standard for the U.S. banking industry. The Australian key management standard, AS2805.6.5.3, also specifies RSA.

View All Answers

**Question - 40:**

Is RSA Currently in Use Today?

**Ans:**

The use of RSA is practically ubiquitous today. It is currently used in a wide variety of products, platforms, and industries around the world. It is found in many commercial software products and is planned to be in many more. It is built into current operating systems by Microsoft, Apple, Sun, and Novell. In hardware, RSA can be found in secure telephones, on Ethernet network cards, and on smart cards. In addition, RSA is incorporated into all of the major protocols for secure Internet communications, including SSL, S-HTTP, SEPP, S/MIME, S/WAN, STT and PCT. It is also used internally in many institutions, including branches of the U.S. government, major corporations, national laboratories, and universities.

View All Answers

**Question - 41:**

What are the Alternatives to RSA?

**Ans:**

Many other public-key cryptosystems have been proposed, as a look through the proceedings of the annual Crypto, Eurocrypt, and Asiacrypt conferences quickly reveals. Some of the public-key cryptosystems will be discussed in previous Question.
A mathematical problem called the knapsack problem was the basis for several systems, but these have lost favor because several versions were broken. Another system, designed by ElGamal, is based on the discrete logarithm problem. The ElGamal system was, in part, the basis for several later signature methods, including one by Schnorr [Sch90], which in turn was the basis for DSS, the Digital Signature Standard. The ElGamal system has been used successfully in applications; it is slower for encryption and verification than RSA and its signatures are larger than RSA signatures.
In 1976, before RSA, Diffie and Hellman proposed a system for key exchange only; it permits secure exchange of keys in an otherwise conventional secret-key system. This system is in use today.

View All Answers

**Question - 42:**

How is RSA Used for Authentication in Practice? What are RSA Digital Signatures?

**Ans:**

RSA is usually combined with a hash function to sign a message.
Suppose Alice wishes to send a signed message to Bob. She applies a hash function to the message to create a message digest, which serves as a "digital fingerprint" of the message. She then encrypts the message digest with her RSA private key; this is the digital signature, which she sends to Bob along with the message itself. Bob, upon receiving the message and signature, decrypts the signature with Alice's public key to recover the message digest. He then hashes the message with the same hash function Alice used and compares the result to the message digest decrypted from the signature. If they are exactly equal, the signature has been successfully verified and he can be confident that the message did indeed come from Alice. If they are not equal, then the message either originated elsewhere or was altered after it was signed, and he rejects the message. With the method just described, anybody read the message and verify the signature. This may not be applicable to situations where Alice wishes to retain the secrecy of the document. In this case she may wish to sign the document then encrypt it using Bob's public key. Bob will then need to decrypt using his private key and verify the signature on the recovered message using Alice's public key. A third party can also verify the signature at this stage.

View All Answers

**Question - 43:**

How do You Know if a Number is Prime?

**Ans:**

It is generally recommended to use probabilistic primality testing, which is much quicker than actually proving that a number is prime. One can use a probabilistic test that determines whether a number is prime with arbitrarily small probability of error, say, less than 2-100.

View All Answers

**Question - 44:**

Can Users of RSA run out of Distinct Primes?

**Ans:**

There are enough prime numbers that RSA users will never run out of them. The Prime Number Theorem states that the number of primes less than or equal to n is asymptotically n/log n. This means that the number of prime numbers of length 512 bits or less is about 10150, which is a number greater than the number of atoms in the known universe.

View All Answers

**Question - 45:**

How Large Should the Primes be?

**Ans:**

The two primes, p and q, which compose the modulus, should be of roughly equal length; this will make the modulus harder to factor than if one of the primes was very small. Thus if one chooses to use a 768-bit modulus, the primes should each have length approximately 384 bits. If the two primes are extremely close (identical except for, say, 100 - 200 bits), there is a potential security risk, but the probability that two randomly chosen primes are so close is negligible.

View All Answers

**Question - 46:**

How Large a Modulus (Key) Should be Used in RSA?

**Ans:**

The best size for an RSA modulus depends on one's security needs. The larger the modulus, the greater the security, but also the slower the RSA operations. One should choose a modulus length upon consideration, first, of one's security needs, such as the value of the protected data and how long it needs to be protected, and, second, of how powerful one's potential enemies are.
Odlyzko's paper considers the security of RSA key sizes based on factoring techniques available in 1995 and the ability to tap large computational resources via computer networks. A specific assessment of the security of 512-bit RSA keys shows that one may be factored for less than $1,000,000 in cost and eight months of effort in 1997 [Rob95d]. It is believed that 512-bit keys no longer provide sufficient security with the advent of new factoring algorithms and distributed computing. Such keys should not be used after 1997 or 1998. Recommended key sizes are now 768 bits for personal use, 1024 bits for corporate use, and 2048 bits for extremely valuable keys like the key pair of a certifying authority. A 768-bit key is expected to be secure until at least the year 2004.

View All Answers

**Question - 47:**

Are Strong Primes Necessary in RSA?

**Ans:**

In the literature pertaining to RSA, it has often been suggested that in choosing a key pair, one should use so-called "strong" primes p and q to generate the modulus n. Strong primes are those with certain properties that make the product n hard to factor by specific factoring methods; such properties have included, for example, the existence of a large prime factor of p-1 and a large prime factor of p+1. The reason for these concerns is that some factoring methods are especially suited to primes p such that p -1 or p+1 has only small factors; strong primes are resistant to these attacks.

View All Answers

**Question - 48:**

What Would it Take to Break RSA?

**Ans:**

There are a few possible interpretations of "breaking RSA." The most damaging would be for an attacker to discover the private key corresponding to a given public key; this would enable the attacker both to read all messages encrypted with the public key and to forge signatures. The obvious way to do this attack is to factor the public modulus, n, into its two prime factors, p and q. From p, q, and e, the public exponent, the attacker can easily get d, the private exponent. The hard part is factoring n; the security of RSA depends on factoring being difficult. In fact, the task of recovering the private key is equivalent to the task of factoring the modulus: you can use d to factor n, as well as use the factorization of n to find d. It should be noted that hardware improvements alone will not weaken RSA, as long as appropriate key lengths are used; in fact, hardware improvements should increase the security of RSA.

View All Answers

**Question - 49:**

How Fast is RSA?

**Ans:**

An "RSA operation," whether for encrypting or decrypting, signing or verifying, is essentially a modular exponentiation, which can be performed by a series of modular multiplications.
In practical applications, it is common to choose a small public exponent for the public key; in fact, entire groups of users can use the same public exponent, each with a different modulus. (There are some restrictions on the prime factors of the modulus when the public exponent is fixed.) This makes encryption faster than decryption and verification faster than signing. With typical modular exponentiation algorithms, public-key operations take O(k2) steps, private-key operations take O( k3) steps, and key generation takes O(k4) steps, where k is the number of bits in the modulus. ( O-notation refers to the upper bound on the asymptotic running time of an algorithm.) "Fast multiplication" techniques, such as FFT-based methods, require asymptotically fewer steps, though in practice they are not as common due to their great software complexity and the fact that they may actually be slower for typical key sizes.

View All Answers

**Question - 50:**

What is RSA?

**Ans:**

RSA is a public-key cryptosystem for both encryption and authentication; it was invented in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman [RSA78]. It works as follows: take two large primes, p and q, and find their product n = pq ; n is called the modulus. Choose a number, e, less than n and relatively prime to (p-1)(q-1), which means that e and (p-1)(q-1) have no common factors except 1. Find another number d such that (ed - 1) is divisible by (p-1)(q-1). The values e and d are called the public and private exponents, respectively. The public key is the pair (n,e); the private key is (n,d). The factors p and q maybe kept with the private key, or destroyed.

View All Answers

# Cryptography Most Popular & Related Interview Guides

1 : **Typesetter Interview Questions and Answers.**

**2 : Ciphers Interview Questions and Answers.**

**3 : Cryptography Teacher Interview Questions and Answers.**

**4 : Typewriter Interview Questions and Answers.**

**5 : Cryptography Interview Questions and Answers.**

**6 : Encryption Decryption Interview Questions and Answers.**

**7 : Digital Certificates Interview Questions and Answers.**

**8 : Cryptography Algorithm Interview Questions and Answers.**

**9 : Cryptography Protocols Interview Questions and Answers.**

**Follow us on FaceBook**
**www.facebook.com/InterviewQuestionsAnswers.Org**

**Follow us on Twitter**
**https://twitter.com/InterviewQA**

**For any inquiry please do not hesitate to contact us.**

**Interview Questions Answers.ORG Team**
**https://InterviewQuestionsAnswers.ORG/**
**support@InterviewQuestionsAnswers.ORG**