

Current State and Future Trend of Encryption Technology

Author: Mitsuru Matsui*

1. Introduction

A surprisingly long time has passed since encryption technology first came into use in our own backyard. Now, ciphers are used in many of the items that we carry around, such as cellular phones, cash cards, train tickets, mobile PCs, and electronic car keys. We live in an age where almost everyone uses encryption in their daily lives, often without even realizing it.

Looking back over the development of encryption technology, we can see that the purpose of using ciphers has extended from “confidentiality” for protecting confidential information to “authentication” for preventing spoofing and “integrity” for preventing forgery of information; and that today’s processors and devices with lower power consumption and higher speed have greatly increased the potential for cryptographic application.

Meanwhile, the advances made in encryption technology also mean advances in cryptanalysis technology. The scientific community is presently engaged in research on cryptanalysis with the goal of identifying any problems in the current encryption systems. The consensus of encryption researchers is that advances made in cryptanalysis technology will mean advances in encryption technology.

Academia requires that encryption algorithms have universal and extremely high security. Therefore, the relationship between cryptanalysis in an academic sense and cryptanalysis in a specific practical application is not always self-evident. Serving as a bridge between them is an important role of encryption researchers in companies.

Recently, researchers pointed out that several widely used encryption systems are at risk of being compromised or deteriorating in security, which could affect the use of encryption from 2010 onward. This is the so-called “year 2010 encryption problem” under global discussion.

Against this backdrop, this paper focuses on the encryption systems currently used in our own backyard and discusses the present state of security evaluation from a practical viewpoint. Also presented is an overview of new encryption technologies currently attracting attention, and a summary of their significance from the viewpoint of convenience and security.

2. Security of Hash Functions

2.1 Hash Function

The hash function is a cryptographic component that finds a great number of applications such as for encrypted passwords, digital signatures, and random number generation. It is also referred to as a cryptographic hash function to clarify the use of encryption.

The role of the hash function is to receive data of an arbitrary length as input and “compress” it to generate a hash value of fixed length. Important security requirements for the hash function include one-wayness and collision resistance. One-wayness means that it is difficult to compute the input backwards from the output, and collision resistance means that it is difficult to detect two different input messages where the hash values are the same.

Figure 1 shows an example of using the hash function in a digital signature. A signer uses the hash

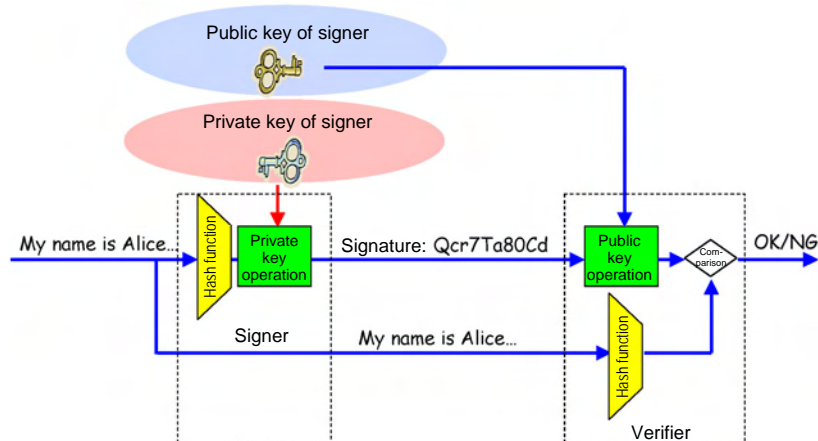


Fig. 1 Example of using the hash function in a digital signature

function to process a message and then perform private key operation. In this case, the collision of hash functions can be disastrous. That is, the capability to compute two different messages, M and M', where Hash(M) = Hash(M') refers to the impossibility of the verifier being able to distinguish digital signature M from M.' Thus, collision resistance is a vital property of the hash function.

2.2 Examples of hash functions

The current most widely used hash function is U.S. Government Standard SHA-1⁽¹⁾. Until SHA-1 was established, the MD series hash functions, MD4 and MD5, were in wide use. Even now, some applications use MD5 for the purpose of compatibility. However, as described in this section, the use of the MD series hash functions cannot be recommended in terms of security.

The hash length of SHA-1 is 160 bits. A newly established U.S. Government Standard, SHA-2, has a longer hash length. SHA-2 is the generic name for several hash functions, and individual algorithms are called SHA-256, SHA-384, and SHA-512 using the format in which the hash length follows the name.

2.3 Security problem of hash functions

It is generally known that the collision of hash functions with a hash length of n bits can be found by a computation amount of 2^{n/2}. Thus, the upper limit of hash function security becomes 2^{n/2}. However, it was recently found that the collision of some hash functions can be obtained by a computation amount that is smaller than 2^{n/2}. Examples of actual collision of MD4 and MD5 in particular have been reported⁽²⁾. Although the collision of SHA-1 has not yet been reported, discovery is expected within one or two years.

Table 1 is a summary of the present state of security of hash functions. The step count here indicates the iteration count of the internal basic functions in the hash function. Since MD4 and MD5 collisions are easily found and the possibility of spoofing and forgery may

actually arise in some applications, care must be taken and it is recommended that these hash functions not be used whenever possible.

The computation amount for obtaining collision of SHA-1 is reported to be about 2⁶³⁽³⁾, which is one hundred thousandth of its proper computation amount of 2⁸⁰. For SHA-0, the initial version of SHA-1, although its algorithm document differs in only one line from that of SHA-1 (specifically, SHA-1 requires one more rotation shift operation compared to SHA-0), it is already in a state in which collision is easily found. When this fact is taken into consideration, the risk of SHA-1 being compromised cannot be disregarded.

For SHA-2, the likelihood of the collision of algorithms being found has not yet been reported, and it is considered that there are no problems with SHA-2 even in an academic sense.

2.4 Future of hash functions

The U.S. National Institute of Standards and Technology (NIST), which establishes the Federal Information Processing Standards, recently announced that the use of SHA-1 for the purpose of digital signatures will be discontinued in 2010, and, consequently, a rapid shift from SHA-1 to SHA-2 is currently taking place. Even though SHA-2 does not presently pose a security problem, its structure is similar to that of SHA-1, and thus it is considered that hash functions with a new structure should eventually be standardized.

For this reason, NIST initiated a project in 2008 to select a government standard encryption ASH (Advanced Hash Standard) after SHA-2, and a new standard is expected around 2012. Therefore, the use of SHA-2 will continue only until the new standard is established.

There is almost no application where its security is threatened because of the discovery of one SHA-1 collision, but the shift to a new hash function is required in systems where its validity must be guaranteed for a long time, such as for digital signatures.

Thus, for actual systems, the shift to a new hash function should be determined according to whether the system is affected by the recently discovered hash function collisions and the validity period of the system and data.

3. Security of Public Key Encryption

3.1 RSA encryption

Developed in the mid-1970s, RSA encryption is one of oldest and most widely used public key encryption technologies. Its security is based on the difficulty of a problem involving factorization into prime numbers, and if the key length (composite length) is increased, cryptanalysis should become exponentially difficult (factorization into prime numbers becomes difficult).

Table 1 Present state of security of hash functions

	Hash length	Block length	Step count	Standard	Collision
MD4	128 bit	512 bit	48	RFC1320	×
MD5	128 bit	512 bit	64	RFC1321	×
SHA0	160 bit	512 bit	80		×
SHA1	160 bit	512 bit	80	FIPS180-2 ISO10118	△
SHA256	256 bit	512 bit	64	Same as the above	○
SHA512	512 bit	1024 bit	80	Same as the above	○

×: Many collisions have already been discovered.
 △: High possibility that a collision will be discovered in the near future.
 ○: There is no indication that a collision will be discovered.

RSA encryption generally uses a key length of 1,024 bits, but 2,048 bits or more may be used in a particularly important system. The computation amount required for 1,024-bit composite factorization into prime numbers is about 2^{80} , or the same degree as the common key encryption of an 80-bit key and the security of a hash function with a hash length of 160 bits.

In RSA encryption, the computation amount required for encryption and decryption is proportional to the cube of the key length. That is, when the key length is doubled, the operation amount octuples. RSA encryption requires the generation of prime numbers each time a key is generated, and this computation amount is generally proportional to the fourth power of the key length. Thus, in an application requiring speed and a system that issues a large amount of digital certificates, the key length significantly affects the entire performance.

3.2 Present state of security of RSA encryption

The lower limit of the computation amount required for factorization into prime numbers is not known and even the fact that it is necessary for exponential time is not mathematically proven. However, factorization into prime numbers is a long-standing mathematical problem, and from the previous research findings, it is considered that factorization into prime numbers cannot be executed in polynomial time.

This indicates asymptotical evaluation or that the security increases dramatically when the key is lengthened, and does not guarantee the computation time of factorization into prime numbers when the key is fixed to a certain length. To determine the potential for individual composite factorization into prime numbers, researchers are continuously running experiments with multiple computers. As listed in Table 2, records have been broken one after another, with the current world record at 640 bits⁽⁴⁾.

Table 2 History of world records for factorization into prime numbers

Number of bits of composite	Date that factorization into prime numbers was announced
430	April 10, 1996
463	February 2, 1999
512	August 22, 1999
530	April 1, 2003
576	December 3, 2003
633	May 9, 2005
640	November 2, 2005

From these results, it is difficult to predict when a 1,024-bit composite would be subjected to factorization into prime numbers, but it is considered that it will be

about 2015 at the earliest. Thus, under the present circumstances, as with the hash function, it is recommended that the key length for RSA encryption be shifted to 2,048 bits for applications requiring long-term evidence to be secured, such as for digital signatures.

However, as previously described, it is not always easy to shift the key length for RSA encryption, since doubling the length significantly affects the application. As described below, it will become important to use a mechanism that extends the effectiveness of cryptographic processing to cope with this problem.

3.3 Identity-based encryption

Identity-based encryption, in which arbitrary information can be set in a public key, was advocated back in 1984. For many years after that, the specific construction of secure identity-based encryption was an unresolved problem. Eventually, around 2000, a system was invented that is practical and has proven security, possibly the ultimate solution⁽⁵⁾⁽⁶⁾. Since then, active research and development has been underway around the world.

Table 3 lists the relationship between the public key and private key for RSA encryption, the elliptic curve cryptosystem, and identity-based encryption. The public key for RSA encryption and the elliptic curve cryptosystem appear on the left side of the relational expression and the private key for identity-based encryption appears on the left side. The essential advantage of identity-based encryption is that the relationship is one where computation can start with an arbitrary public key.

Table 3 Relationship between public key and private key in public key encryption

	Public key	Private key	Relationship between public key and private key
RSA encryption	N	P, q	$N = p \times q$ (p and q: Prime numbers) → N cannot be set to an arbitrary value.
Elliptic curve cryptosystem	y	x	$y = x \cdot P$ (dot: Multiplication of elliptic curve by scalars) (P: Common public information) → y cannot be set to an arbitrary value.
Identity-based encryption	y	x	$x = s \cdot y$ (dot: Multiplication of elliptic curve by scalars) (s: Secret known only by the center) → y can be set to an arbitrary value (= identity).

While one of the purposes of the digital certificate used in the current PKI framework is to prevent spoofing by guaranteeing the relationship between the public key and its owner, identity-based encryption has a breakthrough feature that prevents spoofing even without a certificate since the owner information is embedded in the public key itself.

Meanwhile, the PKI framework, which was stan-

standardized long ago, has a legal basis, is deeply integrated in our daily life, and the certificate has a revocation mechanism. Therefore, the role of identity-based encryption will depend on future applications; it will not be a rival of PKI. Although the products using identity-based encryption are still low in number, research and development towards higher-speed encryption is also making rapid progress and the application of identity-based encryption to embedded systems is expected.

4. Conclusion

This paper explained the present state of security of encryption algorithms and future prospects for encryption technology while giving specific examples. We have proceeded with a comprehensive approach to encryption technology from the development of block encryption and construction of public key encryption and PKI to participation in standardization. We have also been focusing on encryption implementation and quantum encryption application for a long time.

Approaches to individual technologies described here are discussed in the papers of this June 2009 issue. For details, refer to the relevant papers.

References

- (1) "Secure Hash Standard," FIPS Publication 180-2, NIST (2002).
- (2) X. Wang et al., "Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD," Cryptology ePrint Archive 2004/199 (2004).
- (3) X. Wang et al., "New Collision Search for SHA-1," CRYPTO 2005 Rump Session (2005).
- (4) "RSA-640 is factored," RSA laboratories homepage, <http://www.rsa.com/rsalabs/node.asp?id=2964>
- (5) K. Ohgishi, R. Sakai and M. Kasahara, "Basic consideration on ID key sharing scheme on elliptic curves," ISEC99-57 (1999).
- (6) D. Boneh et al., "Identity-based encryption from the Weil pairing," CRYPTO 2001 (2001).