GM

CRYPTOGRAPHY INDUSTRY STANDARD

OF THE PEOPLE'S REPUBLIC OF CHINA

ICS 35.040
L 80
File No.: 49740-2015

GM/T 0041-2015

Cryptographic test specification for smart card

智能 IC 卡密码检测规范

Issued on: April 1, 2015 **Implemented on: April 1, 2015**

Issued by: State Cryptography Administration

Table of Contents

Cryptographic test specification for smart card

1 Scope

This Standard specifies the test items and methods of smart IC card products.

This Standard is applicable to the cryptographic test of smart IC card products; and can also be used to guide the research and development of smart IC card products. Smart IC card products include, but are not limited to, financial IC card, bus IC card, etc.

2 Normative references

The following documents are essential to the application of this document. For the dated references, only the versions with the dates indicated are applicable to this document. For the undated references, the latest version (including all the amendments) are applicable to this document.

GM/T 0005 Randomness Test Specification

GM/T 0039 Security Test Requirements for Cryptographic Modules

GM/Z 4001 Cryptography terminologies

3 Terms and definitions

What defined in GM/Z 4001, and the following terms and definitions are applicable to this document.

3.1 Symmetric cryptographic algorithm

The cryptographic algorithm which uses the same cryptographic key for encryption and decryption.

3.2 Asymmetric cryptographic algorithm/public key cryptographic algorithm

The cryptographic algorithm which uses different cryptographic keys for encryption and decryption. One key (public key) can be made public; the other key (private key) must be kept secret. It is not computationally feasible to solve the private key from the public key.

b) Key security transport testing;

c) Security state and access permission testing;

d) Application firewall testing.

5.3 Cryptographic key primality test

The primality of RSA key generated by smart IC card shall meet the requirement of large prime number.

5.4 Random number quality test

The randomness of the random number generated by smart IC card shall meet the requirement in GM/T 0005.

5.5 Correctness test of cryptographic algorithm implementation

The correctness test of cryptographic algorithm implementation includes the testing on the following 6 aspects:

a) Correctness testing of block algorithm implementation;

b) Correctness testing of key generation of asymmetric key cryptographic algorithm;

c) Correctness testing of encryption and decryption implementation of asymmetric key cryptographic algorithm;

d) Correctness testing of digital signature and signature authentication of asymmetric key cryptographic algorithm;

e) Correctness testing of hash algorithm implementation;

f) Correctness testing of sequence algorithm.

5.6 Performance test of cryptographic algorithm implementation

The performance test of cryptographic algorithm implementation includes the performance testing on the following 11 aspects:

a) Encryption performance testing of block key cryptographic algorithm;

b) Decryption performance testing of block key cryptographic algorithm;

c) Performance testing of hash algorithm;

d) Encryption performance testing of asymmetric key cryptographic

security state. The target of testing shall return that the security state is not met;

c) After the authentication, OPERATE the document which requires security state. The target of testing shall return a successful operation.

6.2.1.2 Abnormal condition testing

The testing steps are as follows:

a) USE a wrong external authentication key to authenticate. The target of testing shall return unsuccessful operation and prompt the remaining number of authentication. When the remaining number of authentication is zero, the external authentication key locks;

b) USE a wrong external authentication key to authenticate. After the authentication, OPERATE the document which requires security state. The target of testing shall return that the security state is not met;

c) USE a wrong key identifier to do external authentication. The target of testing shall return that the cryptographic key is not found;

d) When there are multiple external authentication keys in a target of testing, successfully AUTHENTICATE external authentication key 1 and OPERATE the document protected by external authentication key 2; the target of testing shall return that the security state is not met.

6.2.2 Internal authentication testing

USE the standard testing data to do internal authentication; the results, which shall be returned by target of testing, shall be consistent with the expected results.

6.2.3 PIN authentication testing

6.2.3.1 Normal condition testing

The testing steps are as follows:

a) USE a correct PIN to authenticate. The target of testing shall return a response of successful authentication;

b) Before the authentication, OPERATE the document which needs to be protected by PIN. The target of testing shall return that the security state is not met;

c) After the authentication, OPERATE the document which needs to be

number;

e) The length of PIN exceeds the specified range; the target of testing shall return unsuccess;

f) For 3 consecutive times, USE wrong cryptographic key to calculate MAC for unlocking operation, the application locks.

6.2.7 Application locking testing

6.2.7.1 Normal condition testing

The testing steps are as follows:

a) USE a correct method to calculate MAC for application locking. The target of testing shall return successful application locking;

b) After the application is temporarily locked, only the commands used to select application, take response data, take random number, apply unlocking can be performed. Otherwise, the target of testing returns that the use condition is not met;

c) After the application is permanently locked, only the commands used to select application, take response data, take random number can be performed. Otherwise, the target of testing returns permanent application locking.

6.2.7.2 Abnormal condition testing

The testing steps are as follows:

a) USE a wrong Lc to calculate MAC for application locking operation. The target of testing shall return security message error;

b) USE a wrong filling method to calculate MAC for application locking operation. The target of testing shall return security message error;

c) USE a wrong cryptographic key to calculate MAC for application locking operation. The target of testing shall return security message error;

d) The non-fetched random number directly calculates MAC for application locking operation. The target of testing shall return the non-fetched random number;

e) Under DDF, USE application locking command, the target of testing shall return that the use condition is not met;

USE non-private key document to open the digital envelope; the target of testing shall return unsuccess.

6.3 COS security mechanism test

6.3.1 Message security transport testing

6.3.1.1 Normal condition testing

Testing of ciphertext mode with MAC:

a) In ciphertext mode with MAC, UPDATE basic documents;

b) By sending MAC, READ the content of ciphertext basic documents;

c) The test authority decrypts the ciphertext which has been read;

d) The decrypted data shall be consistent with the write-in content;

e) USE different data lengths to test.

6.3.1.2 Abnormal condition testing

The testing steps are as follows:

a) USE a wrong Lc to calculate MAC for read-write operation. The target of testing shall return security message error;

b) USE a wrong filling method to calculate MAC for read-write operation. The target of testing shall return security message error;

c) USE a wrong cryptographic key to calculate MAC for read-write operation. The target of testing shall return security message error;

d) The non-fetched random number directly calculates MAC for read-write operation. The target of testing shall return the non-fetched random number;

e) USE the plaintext mode to perform read-write operation. The target of testing shall return document type error;

f) USE the ciphertext mode to perform read-write operation. The target of testing shall return document type error;

g) For 3 consecutive times, USE a wrong cryptographic key to calculate MAC for read-write operation, the application locks.

6.3.2 Key security transport testing

e) USE wrong Lc encrypted data to perform read-write operation. The target of testing shall return that the security message data item is not correct;

f) USE ciphertext mode to write the cryptographic key which requires to be written in ciphertext with MAC; the target of testing shall return document type error.

6.3.3 Security state and access permission testing

6.3.3.1 Write document permission testing

The testing steps are as follows:

a) Without getting permission, WRITE document; the target of testing shall return that the security state is not met;

b) After getting permission, to write the document is successfully executed;

c) RE-SELECT the document directory, READ the write-in document; and CONFIRM that the content of the write-in document is correct.

6.3.3.2 Read document permission testing

The testing steps are as follows:

a) Without getting permission, READ document; the target of testing shall return that the security state is not met;

b) After getting permission, to read the document is successfully executed.

6.3.3.3 Write key permission testing

The testing steps are as follows:

a) Without getting permission, WRITE key; the target of testing shall return that the security state is not met;

b) After getting permission, to write the key is successfully executed.

6.3.3.4 Secret key use permission testing

The testing steps are as follows:

a) Without getting permission, USE the key; the target of testing shall return that the security state is not met;

b) After getting permission, the key can be used.

2) OPERATE other locked applications; it shall return that the security state is not met.

e) Document update testing:

1) SELECT Application 1; READ all document contents;

2) SELECT Application 2; UPDATE one of the documents;

3) RETURN to Application 1; READ all document contents. The contents are unchanged.

6.4 RSA key primality test

6.4.1 Prime number collection

USE the command of “prime number generation”; continuously COLLECT *N* prime number pairs. *N* shall not be less than 1000.

6.4.2 Data analysis

VERIFY that the obtained prime number pair data shall meet the primality requirement.

6.5 Random number quality test

For the smart IC card with random number generation function, to ensure the quality of random number, it shall perform this testing.

6.5.1 Random number collection

USE the command of “random number generation”; continuously COLLECT *N* random number documents. *N* shall not be less than 1000. A single document shall not be less than 128 k bytes.

6.5.2 Data analysis

SEE GM/T 0005 for testing method.

6.6 Correctness test of cryptographic algorithm implementation

6.6.1 Correctness testing of encryption and decryption implementation of block cryptographic algorithm

The testing steps are as follows:

a) EXECUTE the operation command of block cryptographic algorithm; USE the specified cryptographic key to perform the operation;

d) CALCULATE signature rate.

6.7.7 Performance testing of signature authentication of asymmetric key cryptographic algorithm

The testing steps are as follows:

a) USE the testing key and standard data to execute the signature authentication command of asymmetric algorithm to perform the operation for 1000 times;

b) VERIFY the correctness of the results;

c) ACCUMULATE the total operation time *T*;

d) CALCULATE signature authentication rate.

6.7.8 Performance testing of key pair generation of asymmetric key cryptographic algorithm

The testing steps are as follows:

a) Continuously GENERATE asymmetric key pairs; PERFORM 1000 operations;

b) ACCUMULATE the total operation time *T*;

c) CALCULATE key pair generation performance.

6.7.9 Encryption performance testing of sequence algorithm

The testing steps are as follows:

a) USE random data and random key to execute the sequence algorithm command to perform the encryption operation for 1000 times;

b) VERIFY the correctness of the encrypted results;

b) ACCUMULATE the total operation time *T*;

c) CALCULATE encryption rate.

6.7.10 Decryption performance testing of sequence algorithm

The testing steps are as follows:

a) USE random data and random key to execute the sequence algorithm command to perform the decryption operation for 1000 times;

This is an excerpt of the PDF (Some pages are marked off intentionally)

------ The End ------