# A New Digital Signature and Certificate Architecture with Shortest Certification Path

Sung Keun Song[1], Hee Yong Youn[1], Kang Shin Lee[2]

[1] School of Information and Communication Engineering
Sungkyunkwan University, 440-746, Suwon, Korea
kkskk103@skku.edu, youn@ece.skku.ac.kr
[2] Critical Information Infrastructure Protection Division
Korea Information Security Agency, 138-803, Seoul, Korea
kslee@kisa.or.kr

**Abstract.** The existing certificate architecture has two problems in terms of security and authentication. One is that there exists some possibility of certificate forgery by exploiting the collision problem associated with the hash algorithm used for signing. The other is that certification path complicates user authentication because it increases according to the distance between communicating users. In this paper we propose a new digital signature scheme and certificate architecture that solve the problems. It is achieved by using two-public key cryptography and a new certificate architecture. The proposed approach can be used without reconstructing the structure of the existing PKI system and always allows a certification path whose length is 1 regardless of the distance between the users. This is because each user confirms only the digital signature of root CA in the combined hierarchical-network infrastructure.

## 1 Introduction

Recently, user authentication has become an important issue in communication and e-commerce. The present target in authentication is implementation in PKI (Public Key Infrastructure) environment so that illegal connection and information leak can be prevented. Here the public key certificate of user is a crucial part, while the main concerns with the certificate are security and certification path. If a certificate is counterfeited by a malicious third party, catastrophic results may occur in the system. Also, the longer a certification path becomes, the larger authentication time and inconvenience to users are [1-4].

A public key certificate is based on digital signature. A certificate is known to the public after signed by a certificate authority (CA). The existing certificate architecture has two problems in terms of security and authentication. One is that possibility of certificate forgery exists. The other is that certification path complicates user authen-

This work was supported in part by 21C Frontier Ubiquitous Computing and Networking, Korea Research Foundation Grant (KRF - 2003 - 041 - D20421) and the Brain Korea 21 Project in 2003. Corresponding author: Hee Yong Youn

tication. The reason why forgery of a certificate is possible is forgery of a digital signature of the certificate is possible. Digital signature algorithms have the collision problem of hash algorithm used for signing. Even though it is not easy for an adversary to attack a certificate by taking advantage of the hash collision problem, it is still possible that an adversary counterfeits the certificate. Here, an important problem is that if a certificate is counterfeited, one cannot prove illegality of the counterfeited certificate except the CA published the certificate. This hash collision problem may cause a devastating result especially when the counterfeited certificate is used for some important services. It is also a pivotal point to optimize the certification path of a certificate in PKI. A certification path is decided according to the distance between communicating users and user authentication depends on it in the existing PKI architecture; hierarchy, network, combined hierarchy-network. Therefore, the longer the certification path becomes, the more user authentication is complicated [1-10].

In this paper we develop the methods solving these problems by proposing a new digital signature scheme and certificate architecture using it. It is based on two-public key cryptography, and the proposed approach can be employed without reconstructing the structure of the existing digital signature scheme. Therefore, we can flexibly select the new scheme or the existing one according to the required degree of security. The proposed certificate architecture always allows a certification path whose length is 1 regardless of the distance between the users because each user confirms only the digital signature of root CA in the combined hierarchical-network infrastructure. Therefore, user authentication can be finished quickly. We provide detail of the proposed signature scheme and analyze its security.

The rest of the paper is organized as follows. Section 2 presents a brief description of digital signature and PKI architecture. Section 3 investigates the vulnerability of digital signature due to collision problem of the hash algorithm and fragile certification path of current PKI architecture. Section 4 proposes a new digital signature scheme and certificate architecture, and security of the scheme is evaluated. Finally, we conclude the paper in Section 5.

## 2 The PKI Architecture

### 2.1 Digital Signature

A digital signature is a pair of large numbers represented as strings of binary digits. Digital signature is computed using a set of rules and parameters with which identity of the signatory and integrity of the data can be verified. An algorithm is used to provide the way how to generate and verify the signature. The signature generation process makes use of a private key to generate a digital signature, while the signature verification process makes use of a public key corresponding to the private key. Each user possesses a private and public key pair. Public keys are known to the public using the certification of CA in general. Private keys are never shared. One can verify the signature of a user by using the user’s public key. Only the possessor of a private key can generate signatures as long as the key has not been revealed [1].

A hash algorithm is used in the signature generation process to obtain a condensed version of message, called a message digest. The message digest is then input to the digital signature algorithm to generate a digital signature. The digital signature is sent to the intended verifier along with the message. The verifier of the message and signature verifies the signature using the sender's public key [1].

The same hash algorithm as the one used by the sender must be used in the verification process. The hash algorithm is specified in a separate standard, the Secure Hash Standard, FIPS 180-1 [2]. FIPS approved several digital signature algorithms implemented with the Secure Hash Standard. Similar procedures may be used to generate and verify signatures for stored as well as transmitted data.

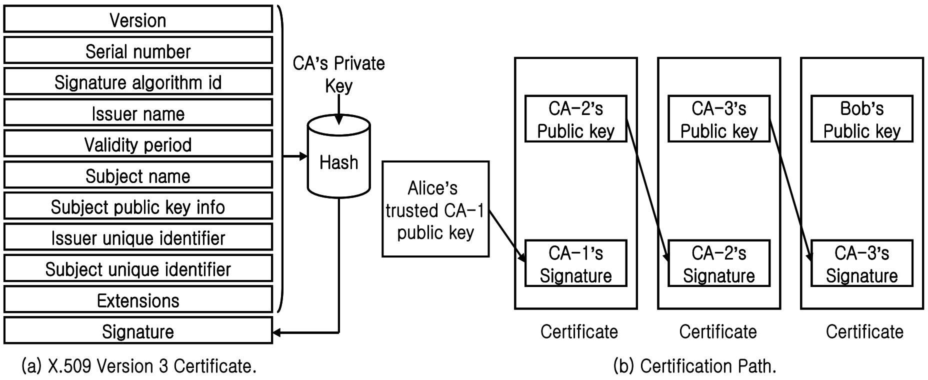


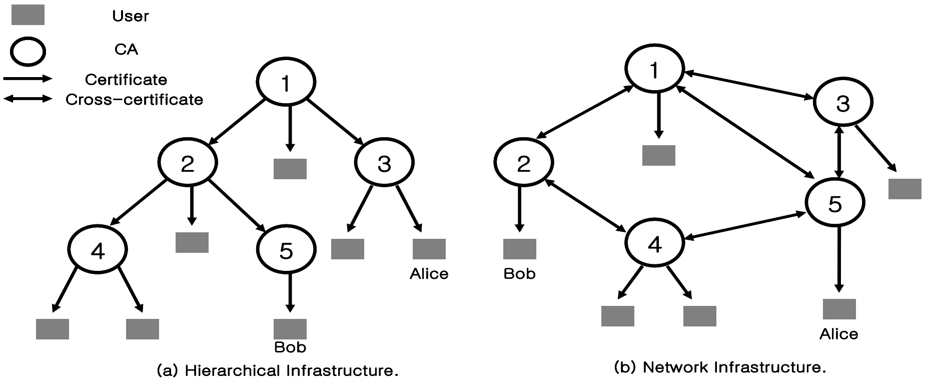
Figure 1. Certificate architecture and certification path.

## 2.2 PKI architecture

Figure 1(a) illustrates the X.509 v3 certificate. A certificate includes the issuer name, the subject name and the subject's public key, and is signed with the issuer's private key. For example, if Alice has Bob's certificate and knows the issuing CA's public key, she can verify Bob's certificate and then use Bob's public key to verify Bob's signature in any document. Certification path is a chain of certificates that use trust relationship between the CAs to determine if a certificate signed by a CA is trusted. This is illustrated in Figure 1(b); Bob has been issued a certificate by CA-3, which has been issued a certificate by CA-2, which in turn has been issued a certificate by CA-1. If Alice trusts CA-1 and knows its public key, she can verify each certificate in the certification path until she reaches Bob's certificate and verifies it. At that point, Alice knows Bob's public key and can verify his signature. CAs can certify each other in a systematic manner to form a PKI. A CA may be issued a certificate by another CA. Two CAs may issue each other a certificate; this is known as cross-certification, and the pair together is a cross-certificate.

PKI architectures fall into three configurations: hierarchy, network, combined hierarchy-network. Each configuration is characterized by the number of CAs, the trust relationship between the CAs, and where the PKI users place their trusts [1-4].

### 2.2.1 Hierarchical PKI



(a) Hierarchical Infrastructure. (b) Network Infrastructure.

Figure 2. PKI architecture.

PKIs constructed with superior-subordinate CA relationships are called hierarchical PKIs. The foundation of such an architecture is the "root" CA, which issues certificates to subordinate CAs but not to users. Subordinate CAs, in turn, issue certificates to users or other subordinate CAs located below them in the hierarchy. In hierarchical PKIs, the trust relationship is one-directional; subordinate CAs do not issue certificates to their superior CAs. Figure 2(a) shows an example of a hierarchical PKI with the root CA numbered as 1. The superior CAs impose rules governing the types of certificates their subordinates can issue; applications leverage these conditions to identify acceptable certificates. Since certificate suitability is implied by issuing the CA's identity, application-specific information can be omitted from the certificates. Every user knows the public key of the root CA, and any user's certificate may be verified by verifying the certification path that leads back to the root CA. For example, Alice verifies Bob's certificate, issued by CA-5, then CA-5's certificate issued by CA-2, and then CA-2's certificate issued by CA-1, the root, whose public key she knows.

The hierarchical PKI architecture has some advantages. The structure of many organizations such as government is largely hierarchical and trust relationships are frequently aligned with the organizational structure. A hierarchical PKI may be aligned with hierarchical directory names, and the search strategy of certification path is straightforward. Each user has a certification path back to the root; the user can provide this path to other users and every user can verify the path since all users know the root's public key.

Meanwhile, the hierarchical certification path architecture has some disadvantages. It is improbable that there will be a single root CA in the world, and therefore cross-certificates must exist at some level. Also, certification path verifiers must be able to cope with the topologies that are not entirely hierarchical. Moreover, compromise of the root private key is catastrophic because every certification path is compromised and recovery requires secure "out-of-band" distribution of the new public key to every user [1-4].

### 2.2.2 Network PKI

The traditional alternative to hierarchical PKIs is to create a network PKI or web of trust to connect CAs via P2P relationships. A CA in a network PKI can be a trust anchor, although users generally consider the CA issued their certificates as their trust anchor. In this architecture, CAs issue certificates to each other, and a pair of certificates describes a bi-directional trust relationship. Specific P2P CAs can specify any limitation of trust in the certificates they exchange between them. Figure 2(b) illustrates a network PKI. A user knows the public key of a CA near itself, generally the local CA that issued its certificate, and verifies the certificates by verifying the certification path leading back to the trusted CA. For example, Alice knows the public key of CA-5. There are several certification paths that lead from Bob to Alice, but the shortest path requires Alice to verify Bob's certificate issued by CA-2, then CA-2's certificate issued by CA-4, and finally CA-4's certificate issued by CA-5. CA-5 is Alice's CA, and she trusts CA-5 and knows its public key.

The network PKI architecture has the advantages that it is flexible and facilitates adhoc associations and trust relationships, and readily reflects bilateral trust relationships. It is likely that a national or worldwide PKI will evolve in an adhoc fashion involving isolated CAs, and this can be more easily accommodated in a network than a hierarchy. The CAs widely spread out but supporting the users working together with a high degree of trust can be directly cross-certified under a high trust policy that is higher than would be practical through a long, hierarchical chain of certificates. The CAs whose users communicate frequently can cross-certify directly, which can reduce certification path processing.

Perhaps the most compelling argument for a network PKI is that it is more convenient and natural for a certificate holder to place its trust in the local CA issued its certificate rather than a remote root CA, and make this the foundation of all trust relationships. Moreover, this simplifies the out-of-band secure distribution of the public key of CA. Also, recovery from the compromise of any CA's private key requires only that the new public key is securely distributed to the holders of the certificates issued from that CA and new certificates are generated for them. The network PKI has at least two disadvantages though [1-4]:

- Search of efficient certification path is complex.
- A user cannot provide a single certification path guaranteeing verification of its signatures by all other users of the PKI.

### 2.2.3 Combined Hierarchical-Network Federal PKI

The hierarchical and network PKI architecture are not mutually exclusive. Figure 3 illustrates a combined hierarchical-network federal PKI. There is a hierarchical path of certificates leading from the root CA to its subordinate CAs, and from each of these CAs to their subordinates, and so on, until every Federal end user is issued a certificate with a certification path from the root CA. Each Federal CA will have a single parent. There is one or more instances of the directory attribute certificate for the certificates issued by the parent. There is only one hierarchical path to the root

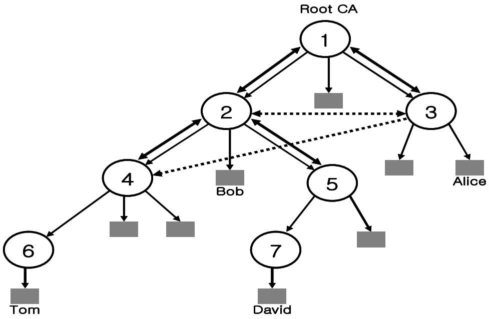


Figure 3. Combined hierarchical-network architecture.

CA based on the directory attribute certificate. Other certificates held by a CA, which was generated by another issuer, will be posted in the directory in a cross-certificate pair. In parallel to the certificates hierarchically linking CAs to the root cross-certificate pair attributes also link those CAs. These parallel cross-certificate pairs are shown in Figure 3 as solid double-headed arrows. This allows client applications to operate with any Federal CA that perform certification path verification from the verifier's parent CA using the cross-certificate pair directory attribute. Federal CAs may cross-certify each other along the paths not parallel in the hierarchy. Optional cross-certificate pairs are shown in Figure 3 as dotted double-headed arrows. If Alice wishes to verify Bob's signature, she can find either a certification path that relies on her trust in her parent CA, CA-3, or Bob's certification path back to the root. In general, Federal PKI clients and applications may choose to follow either a certification path verification strategy that leads to the root CA, or back to their own CA. Because of the hierarchical cross-certificates, a certification path is guaranteed to exist from a client's own CA to every Federal certificate through the root CA, but there may also be much shorter paths [2].

## 3 Vulnerabilities of PKI

### 3.1 Digital Signature

As explained earlier, digital signature algorithms have two connoted hazardous factors in terms of security. They are the inherent security limitation of a digital signature algorithm and the collision problem of hash algorithm used for signing. The security of a digital signature algorithm depends on the security of public key cryptography. The collision problem of the hash algorithm, the second hazardous factor, is another factor limiting the security of digital signature.

A hash algorithm maps an arbitrary-length message to a fixed-length hash value, which must be a fast operation. On the other hand, the hash algorithm must be collision-resistant, i.e. it must be computationally infeasible to find a collision, which is a pair of different messages with the same hash value. However, collision cannot be avoided. MD5, SHA, and RIPEMD-160 are representative hash algorithms [7-9].

Many of the existing hash algorithms follow a design principle of Merkle-Damgard [10] shown in Figure 4. Essentially, this model simplifies the management of large inputs and production of a fixed-length output by using a function *F*, which is usually called a compression function. Given a compression function, a hash algorithm can be defined as repeated applications of the function until the entire message has been processed.

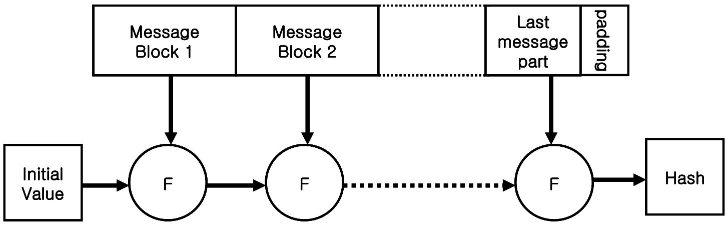


Figure 4. Merkle-Damgard model.

In this process a message of arbitrary length is broken into blocks whose length depends on the compression function, and padded so that the size of the message becomes a multiple of the block size. The blocks are then processed sequentially, taking the result of hashing so far and the current message block as input, with the final output being the hash value for the entire message. The hash function is repeatedly applied to the message block and hash value of the previous blocks. The security of this scheme rests on the security of the *F* function. Note that the more the message size increases, the more the number of collisions per hash value increases exponentially. For example, we assume that one message block is 512 bits and the *F* function returns a 128-bit output. When a message needs 1 block, the number of collision is

$$\frac{1\times 2^{512}}{2^{128}} = 2^{384}$$

When a message needs 2 blocks, the number of collision is

$$\frac{2^{128}\times 2^{384}\times 2^{512}}{2^{128}} = 2^{896}$$

When a message needs 3 blocks, it is

$$\frac{2^{128}\times 2^{896}\times 2^{512}}{2^{128}} = 2^{1408}$$

In general, for a message of *n* blocks,

$$\frac{2^{128}\times 2^{512n\text{-}640}\times 2^{512}}{2^{128}} = 2^{512n\text{-}128}$$

Namely, $2^{512n\text{-}128}$ messages per hash value have a same value. Because of this property, a third party can counterfeit a certificate signed by a CA. It is due to the collision problem of the hash algorithm, and many critical problems may occur in communication and e-commerce if that happens.

We can classify the attacks taking advantage of the collision problem of a hash algorithm into three types.

- An attacker researches the structural weakness of the hash algorithm to identify collision.

- An attacker accumulates digital signatures corresponding to each hash value for the life time of a public key of a target. For counterfeits, the attacker finds a digital signature from the database whose hash value is equal to that of the counterfeited message to a legitimate target message.
- An attacker counterfeits the message by modifying the counterfeited message until the hash value of it becomes same as that of the target message.

Among these types, the second case is the largest for counterfeiting a certificate. This is because a CA issues many certificates and an attacker can easily accumulate digital signatures of the CA.

### 3. 2 Certification Path

As explained in Section 2.2, the existing PKI architectures have a disadvantage in terms of certification path. It is that a certification path increases according to the distance between communicating users. This problem may complicate user authentication and give vexation to the users.

## 4 The Proposed Certificate Architecture

We have explained vulnerability of PKI. This section proposes a new digital signature scheme and certificate architecture solving the problem. The new digital signature scheme uses a cryptographic algorithm employing two different public keys. In this paper we call it “two-public key cryptography”. The basic idea is to hide the hash value of a certificate an issuer signed using the two-public key cryptography. In the case of users the validity of the digital signature of a certificate is confirmed by the digital signature of root CA and public key of root CA. In the case of CAs the validity of the digital signature of a certificate is confirmed by the semipublic key of the issuer. First, we explain the two-public key cryptography. Then, we propose the new digital signature scheme.

### 4. 1 New Digital Signature Scheme

#### 4.1.1 The Two-Public Key Cryptography

Figure 5 shows the structure of the proposed two-public key cryptography. Note that if the private key is used to encrypt something using Algorithm-B, only public key-2 can decrypt it. That is, the public key that can decrypt the message varies according to the algorithm used for encryption.

We show an example of two-public key cryptography using the RSA and the El-Gamal scheme, the two representative public key cryptography algorithms. First, we review the two.

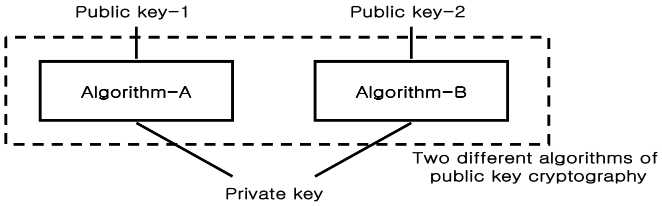


Figure 5. The structure of two public key cryptography.

The RSA cryptography, named after its inventors R. Rivest, A. Shamir, and L. Adleman, is the most widely used public-key cryptography. It may be used to provide both secrecy and digital signatures, and its security is based on the intractability of the integer factorization problem. Each user creates an RSA public key and a corresponding private key. The users do the following [11]:

1. Generate two large random (and distinct) primes $p$ and $q$, each roughly the same size.
2. Compute $n$=$pq$ and $\phi$=($p$-1)($q$-1)
3. Select a random integer $e$, $1< e < \phi$, such that gcd($e$, $\phi$)=1.
4. Use the extended Euclidean algorithm to compute the unique integer $d$, $1< d < \phi$, such that $ed\equiv 1(\mathrm{mod}\ \phi)$.
5. The public key is ($n$, $e$); private key is $d$.

The ElGamal public-key encryption scheme can be viewed as Diffie-Hellman key agreement in the key transfer mode. Its security is based on the intractability of the discrete logarithm problem and the Diffie-Hellman problem. Each user creates a public key and a corresponding private key. The users do the following [12]:

1. Generate a large random prime $p$ and a generator $\alpha$ of the multiplicative group $Z_p$ of the integers modulo $p$.
2. Select a random integer $a$, $1\leq a \leq p$-2, and compute $y= \alpha^a \bmod p$.
3. The public key is ($p$, $\alpha$, $y$) private key is $a$.

In above, we can recognize that if the prime $p$ of the ElGamal and the $\phi$ of the RSA have a same value, the private key of the RSA and ElGamal are same. If the $a$ of the ElGamal is denoted by the $d$ of the RSA, public key-1 is ($n$, $e$), and public key-2 is ($p$, $\alpha$, $y$), and the common private key is $d$ in the proposed two-public key cryptography. In this way, we can construct two-public key cryptography using the RSA and ElGamal scheme. Of course, we can easily construct various two-public key cryptography scheme using any two different public key cryptographies.

### 4.1.2 The New Digital Signature Scheme with the New Certificate Architecture

$P_A$: a public key of algorithm-A; be known to all objects of PKI
$P_B$: a semipublic key of algorithm-B; be known only to the CA's
$P_{AB}^{-1}$: a private key of the two-public key cryptography
$\{\ \} AP_{AB}^{-1}$: encrypt or decrypt the private key using algorithm-A

C: a certificate including a digital signature of a CA
H: a hash function that extends the input regardless of the value
h: a hash function that reduces the input regardless of the extent

The issue process of a certificate consists of two processes; signature generation process and certificate process. Signature generation process is executed by an issuer and certificate process is executed by root CA.

The issue process of a certificate handled by a general CA is as follows. First, the CA calculates a hash value of the random number (RN), H(RN). Here, the extent of the H(RN) has a fixed block size. The CA calculates a hash value where the H value is added to the certificate, h({M}K, H(RN)). When h is calculated, the H value is put on a specific block of the certificate that the CA selected. The CA generates a digital signature by encrypting the h value, the block position, and the random number using algorithm-B and its own private key. Digital signature of the CA is as follows.

Digital signature: {h({M}K, H(RN))||block position||RN} $BP_{CA\text{-}A}^{-1}$

Thereafter, the CA requests certificate signature from root CA by sending the certificate to it. Figure 6(a) shows the signature generation process.

The certificate process handled by root CA is as follows. The root CA searches a semipublic key of CA-A, $P_B$, from a database using the ID of CA-A, and then decrypts the digital signature to obtain the block position and RN. Thereafter, the root CA calculates a hash value, h({M}K, H(RN)), by using the block position and the RN, and then compares the hash value with the h({M}K, H(RN)) value which is part of the decrypted digital signature of CA-A. If the values are same, the root CA calculates a hash value, h({C}K). The root CA generates a digital signature by encrypting h({C}K) by its own private key and algorithm-A for certifying the digital signature of CA-A. After the root CA calculates h(M), the value and digital signature of itself are attached to the certificate of CA-A. The root CA sends the certificate signed by itself to CA-A. As soon as CA-A receives the certificate from root CA, CA-A issues a certificate to the user.

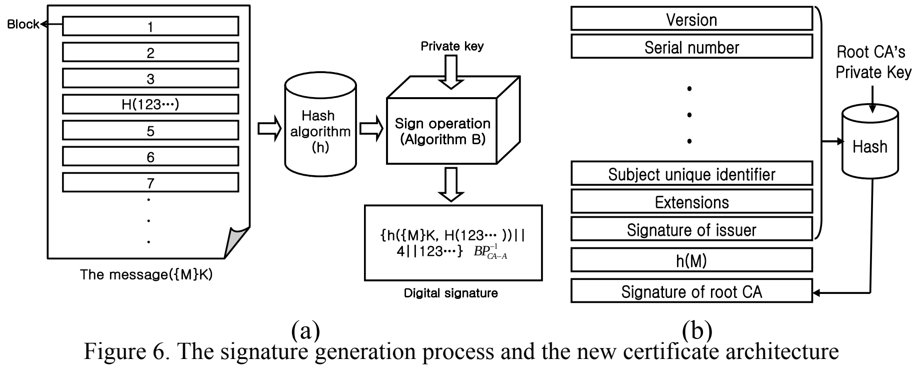


Figure 6. The signature generation process and the new certificate architecture

## 4.2 Optimal Certification Path

The new certificate architecture that we propose is shown in Figure 6(b). The new certificate architecture is obtained by adding h(M) and digital signature of root CA to the existing certificate architecture. The system with the new certificate architecture needs additional certificate of the semipublic key of each CA. Certificates of semipublic keys are commonly used only between the CAs. We assume that all CAs share each certificate of semipublic key of all CAs. Then, all certificates have an optimal certification path regardless of the distance between the users in the combined hierarchical-network infrastructure. For example, in Figure 3, the length of certification path between Tom and David based on the existing certificate architecture is 5 for CA 6 and CA 7 building a cross certificate. If they have a cross certificate, the length of certification path is 2 with the existing certificate architecture. However, whether a cross certification exists or not between CA 6 and CA 7, the length of certification path is always 1 with the new certificate architecture. For all other cases the length is still 1 because each user in the PKI confirms only the hash value, h(M), and the digital signature of  root CA. Therefore, the length of certification path is always 1 with the new certificate architecture. In case a user doubts other's certificate, the user can request verification of the certificate to a CA near itself, generally the parent CA issued its certificate. Then, the CA confirms the certificate using the semipublic key of the issuer of the certificate.

### 4.3 Security of the Proposed Certificate Architecture

The new proposed certificate architecture solves the counterfeit problem presented in Section 3. That is, any user of PKI cannot counterfeit a certificate by taking advantage of the collision problem of the hash algorithm in the new certificate architecture. Without the proposed scheme a malicious user can counterfeit a certificate using the collision problem. With our scheme, a malicious user cannot perfectly counterfeit a certificate since the user is not able to know the RN and block position of the signature of the certificate. The part where forgery is possible in the new certificate architecture by taking advantage of the collision problem of the hash algorithm is the digital signature of root CA. For this reason, the hash value, h(M), is added to the certificate. It is difficult to counterfeit the certificate such that h(M) and h(C) may be valid. However, because of the possibility of forgery, in case a user doubts a certificate, the user must request verification of the certificate to a CA near itself.

We need to compare security of two-public key cryptography and earlier public key cryptography. Note that security of any cryptographic algorithm is influenced by many factors such as difficulty of the mathematical problem of the cryptographic algorithm, complexity of the cryptographic algorithm, and key length, etc. If the securities of the two different cryptographic algorithms employed in the two-public key cryptography are similar, the security of the two-public key cryptography will be similar to the security of each of the two cryptographic algorithms since each of them is based on different problem of mathematics. Therefore, a system designer must design the two-public key cryptography using two different public key cryptographies of the same level of security.

## 5 Conclusion

In this paper we have proposed a new certificate architecture and digital signature scheme solving the collision problem of hashing required in the existing digital signature algorithms. As a result, the security of the new certificate architecture is not limited by the hash algorithm. The new digital signature scheme applied to the new certificate architecture can use a hash algorithm, which allows fast operation while providing high security. The new digital signature scheme can also be used without reconstructing the structure of the existing digital signature scheme.

The new certificate architecture allows an optimal certification path regardless of the distance between the users in the combined hierarchical-network infrastructure. If the e-commerce and communication system share certificates of the semipublic key of all CAs, user authentication can be done effectively with which illegal access is impossible. Therefore, we anticipate that the new certificate architecture can significantly activate e-commerce by increasing the security of transactions and effectively processing user authentication. In the future we plan to investigate the performance of the proposed scheme using various combinations of public key cryptographies.

## References

1. William, T., Nelson E., Polk, Hastings, Ambarish Malpani.: Public Key Infrastructures that Satisfy Security Goals, IEEE Internet Computing. (2003)
2. William, E., Burr, Noel, A., Nazario and W. Timothy Polk.: A Proposed Federal PKI Using X.509 V3 Certificates. NIST. http://csrc.nist.gov/nissc/1996/papers/NISSC96/paper042 /pkipap1.pdf
3. Adams, C., Cain, P., Pinkas, D., Zuccherato, R.: Internet X.509 Public Key Infrastructure Time Stamp Protocol. draft-ietf-pkix-time-stamp-00.txt. (1998)
4. Housely, R., Ford, W., Polk, W., Solo, D.: Internet X.509 Public Key Infrastructure. IETF RFC 2459. (1999)
5. National Institute of Standards and Technology (NIST).: Digital Signature Standard. FIPS PUB 186-2. (2000). http://csrc.nist.gov/publications/fips/fips186-2/fips186-2-change1.pdf
6. National Institute of Standards and Technology (NIST).: Secure Hash Standard. FIPS PUB 180-1. (1995). http://www.itl.nist.gov/fipspubs/fip180-1.htm
7. Dobbertin, H.: The status of MD5 after a recent attack. RSA Laboratories. CryptoBytes, 2(2). (1996)
8. Eastlake, D. 3rd, Jones, P.: US Secure Hash Algorithm 1 (SHA1). RFC 3174. (2001) http://www.faqs.org/rfcs/rfc3174.html
9. Keromytis, A., Provos, N.: The Use of HMAC-RIPEMD-160-96 within ESP and AH. RFC 2857. (2000)
10. Damgard, I.B.: A design principle for hash functions. Advances in Cryptology-Crypto ’89, Lecture Notes in Computer Science, vol. 435, Springer-Verlag, (1990) 416-427
11. Rivest, R., Shamir, A., Adleman, L.: A Method for Obtaining Digital Signatures and Public Key Cryptosystems. Communications of the ACM, (1978) 120-126
12. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Info. Theory, IT-31, No. 4, (1985) 469-472