# Response to Comments on the NIST Proposed Digital Signature Standard

Miles E. Smid
Dennis K. Branstad

National Institute of Standards and Technology[1]

**Abstract.** NIST received comments from 109 separate government agencies, companies, and private individuals concerning the proposed Digital Signature Standard. Both positive and negative comments were received. However the number of negative comments was significantly larger than normally received for a proposed Federal Information Processing Standard (FIPS). This paper summarizes the major comments, both positive and negative, and provides responses where appropriate. The paper highlights the anticipated significant modifications to the proposed standard and concludes by discussing the future milestones that need to be accomplished before the proposed DSS becomes a FIPS.

## 1. Introduction

### 1.1 History of the DSA

In August, 1991 [FRDSS], the National Institute of Standards and Technology (NIST) proposed a Digital Signature Algorithm (DSA) for use in computing and verifying digital signatures in government applications. The DSA was proposed in a draft Digital Signature Standard (DSS) [DFIPSXX] as the initial step of a process leading to a Federal Information Processing Standard.

The goal was to provide a standard for government organizations to use for applications in which a digital signature is required. Private and commercial organizations are encouraged to adopt and use the DSS as well. This paper discusses the primary issues that were raised during the public comment period on the DSS.

The Digital Signature Algorithm is used for mathematically computing and verifying a digital signature. The algorithm explicitly defines the parameters (name, type, size but not value) and specifies the computations for signature generation and verification. A digital signature is simply a number that depends upon the contents of the message and the private key of the message signer. The signature is normally transmitted with the message. A verifier, who has possession of the message, the signature, and the public key of the signer, can determine that the signature was generated by the signer and was not modified, either accidentally or intentionally. In addition, the verifier can provide the message, the digital signature, and the signer's public key as evidence to a third party that the message was, in fact, signed by the claimed signer. Given the evidence, the third party can also verify the signature. This capability is called "nonrepudiation". Of course, one can sign data other than messages, for example, electronic contracts, computer programs, and any valuable electronic information.

### 1.2 Factors Considered

In selecting the Digital Signature Algorithm for the proposed DSS, the following factors were considered important:

---

[1] U.S. Government contribution not subject to copyright.

E.F. Brickell (Ed.): Advances in Cryptology - CRYPTO '92, LNCS 740, pp. 76-88, 1993.

the level of security provided, the applicability of patents, the ease of export from the U.S., the impact on national security and law enforcement, and the efficiency in a number of government and commercial applications. A number of techniques were reviewed and deemed appropriate for providing adequate protection in Federal systems. Among these, NIST placed primary emphasis on selecting the technique that best assures appropriate security for Federal information and does not require payment of royalties by U.S. private or commercial interests. All proposals were coordinated with the national security and law enforcement communities.

A Digital Signature Algorithm should have several technical characteristics. First, it must compute a signature which depends on the contents of the message and the private key of the person that originated it. Second, the private key used for signature generation should not be computable knowing the public key used for signature verification. Third, the efficiency of generating keys, signing messages and verifying messages should have an acceptable impact on performance in various implementations and applications. Fourth, a digital signature algorithm should be useful in many different applications and provide a level of security commensurate with the value or sensitivity of the data being protected.

Several digital signature algorithms have been proposed in the technical literature. Each exhibits the above characteristics to a greater or lesser degree. NIST proposed an algorithm which satisfies the desired technical characteristics in addition to the established non-technical criteria. This paper summarizes the comments received during the first public solicitation for comments on the proposed standard, provides responses to the comments, and discusses planned revisions to the proposed DSS.

### 1.3 GAO Decision B-245714

Government agencies have often raised questions concerning the legality of using a digital rather than a written signature. A "catch 22" condition existed. Agencies would not use digital signature technology because the regulations appeared to require written signatures, and the regulations were not changed or clarified because agencies were not using the new technology. In order to help clarify the issue, NIST requested a formal decision from the General Accounting Office (GAO) [NLET]. Based on its analysis of an agency's financial system and operating procedures, the GAO often grants relief against financial loss. If funds are lost as the result of a weakness in the system or the operational procedures, the loss will come out of general revenues rather than the funds of the agency.

NIST asked the GAO whether NIST standards for electronic signatures could be used to record obligations in government Electronic Data Interchange (EDI) payments. The GAO decision [GAO91] established the criteria for government use of electronic signatures for EDI technologies consistent with 31 U.S.C. Section 1501. Electronic signatures had to be unique and they had to provide a verifiable binding of the individual to the transaction. In particular, the GAO stated that "EDI systems using message authentication codes which follow NIST's Computer Data Authentication Standard (Federal Information Processing Standard (FIPS) 113) or digital signatures following NIST's Digital Signature Standard, as currently proposed, can produce a form of evidence that is acceptable under section 1501."

## 2. Overview of Comments

NIST received comments form 109 separate government agencies, companies, and private individuals concerning the proposed DSS. Both positive and negative comments were received. While government agencies tended to support the proposed standard, the number of negative comments was significantly larger than normally received for a proposed FIPS. The comments are public and copies are available for inspection at the Central Reference and Records Inspection Facility, room 6020, Herbert C. Hoover Building, 14th Street between Pennsylvania and Constitution Avenues, N.W., Washington, DC 20230.

## 3. Sample of Positive Comments

Many responders to the NIST solicitation for comments stated their belief that a digital signature capability will be necessary in electronic funds transfer, electronic data interchange, payroll, and administrative systems. Several

responders supported the government's goal of having a standard that was free of patent impediments and expressed their desire that there be a federal standard for digital signatures which would provide for interoperability and a common level of security. Many government agencies supported the proposed standard. A sample of some positive comments is provided below:

1. The DSA will be especially useful to the financial services industry

2. The DSS is the key to robust and secure transfer of funds between individuals, financial institutions, governments and corporations

3. There will be minimal cost impact if the proposed standard is implemented

4. Generating keys for the DSA is a relatively efficient operation

5. The DSA is the only signature algorithm that has been publicly proposed by any government

6. We recommend that the algorithm be adopted as a FIPS

7. The Department applauds NIST's work in developing a DSS that will help to meet the needs of Federal departments and agencies....

## 4. Response to Negative Comments

Like the Data Encryption Standard (DES) proposed fifteen years earlier as a Federal Information Processing Standard, the DSS received many negative comments, but the comments generally fell into one of several categories. Some responders believed that since the selection process of the proposed DSA had not been public, the usual standards making process was not followed. Other people thought the solicitation for comments was the end of the standards process rather than just the beginning and therefore did not believe sufficient time was being provided for evaluation of the proposal. Many noted that the proposal was an alternative to the Rivest, Shamir and Adleman (RSA) algorithm [RIVEST] that has achieved a high degree of public acceptance. Selecting an alternative to the RSA was felt to have a negative impact by those that had a financial interest and a positive impact by some that had alternative financial interests. Finally, several technical concerns were expressed regarding the security and efficiency of the proposed algorithm. These concerns and responses are summarized below.

### 4.1 The DSA selection process was not public

**Response:**

The early discussions leading to the proposal of the DSA algorithm were not public. The Computer Security Act of 1987 states that NIST "shall draw upon computer system technical security guidelines developed by the National Security Agency ...." [CSA87]. NIST followed its normal standards development procedures, the provisions of the act, and the memorandum of understanding established with the National Security Agency (NSA). Several alternatives were considered before the DSA was selected. The cooperation between NIST and NSA was publicly known. NIST advised the appropriate ANSI accredited standards committees, as well as others, of the joint effort.

In the normal standards development process, NIST identifies the need for a standard, produces technical specifications of a standard using inputs from different sources and then solicits government and public comment on the proposal. After the comment period, the comments are analyzed, appropriate changes are made and a revised standard issued (or further comment is solicited if the revisions are substantial). This public process is being followed. NIST made the specification of the algorithm public and then solicited comments on the proposed algorithm. NIST personnel have given talks on the DSS to Accredited Standards Committee (ASC) X9, Working Group X9F1, Interop '92, the First International Symposium on Cryptographic Security, the Federal Computer

Security Program Managers' Forum, and the NIST Computer Security and Privacy Advisory Board. Working Group X9F1, which makes financial standards related to public key cryptography, is now developing a standard that is equivalent to the DSA [DANSIX9].

### 4.2 Sufficient time for analysis has not been provided

Several parties felt that the three month comment period did not provide sufficient time for analysis of the algorithm. In response to a formal request, NIST extended the comment period for another three months. Few new comments were provided after the initial three month period.

**Response:**

NIST considered the initial three month comment period to be only part of the total DSS evaluation process. The security of the DSA is believed to be equivalent to the difficulty of solving the discrete logarithm problem which has been studied for several years. The ElGamal technique, upon which the DSA is based, has been studied since 1984 and remains basically sound. The DSA does have some new features. In particular, r is calculated by computing $(g^k \bmod p) \bmod q$. However, the new features as well as the entire algorithm were evaluated by the NSA and underwent the same analysis used by NSA to evaluate classified cryptographic systems. In fact, the DSA may be used to sign unclassified data processed by "Warner Amendment" systems (10 U.S.C. 2315 and 44 U.S.C. 3502(2)) as well as classified data in selected applications [FRDSS].

It is now almost a year since the algorithm was publicly proposed and no cryptographic shortcut attacks have been found. NIST will continue to evaluate the merits of any proposed attack and will formally review the DSS at five year intervals. However, to be sure that there is no additional, currently unknown information about the algorithm or its revision (see Section 5.2 below), NIST has stated there will be a second public comment period on a revised DSS proposal before it is published as a standard.

### 4.3 The DSA may infringe on other patents

**Response:**

One of the selection criteria for the DSA was that it be free of patent impediments to the maximum extent possible. An agreement to grant non-exclusive, royalty free licenses had been made by the International Business Machines Corporation in 1975 prior to adopting the DES, which was covered by IBM patents, as a Federal Information Processing Standard. A similar status was desired for the DSS. Some alternative algorithms were considered less desirable because of known patent impediments. The DSS was designed by the government specifically to meet the selection criteria, including the patent criteria. However, two claims of infringement (by Public Key Partners and Professor Claus P. Schnorr) were received during the comment period. In addition other comments expressed a concern that the DSS infringed the patents held by these entities.

A major criterion for the invention and selection of the DSS by the government was to avoid patented technology that could result in payment of royalties for government, commercial and private use. This was stated in Congressional testimony in June, 1991, shortly before the DSS was issued for comment. A patent application was filed for the DSA on behalf of the government with the intent of making the DSS available on a non-exclusive, royalty-free basis. The patent claims were recently allowed by the U.S. patent office. The patents that are claimed to be infringed were directly or indirectly referenced in the DSA patent application.

Based on its initial analysis of existing patents, NIST believed the DSA did not infringe on any known patents. As a result of the claims of infringement, NIST is attempting to clarify the patent issue (see Section 5.1). The judgment of infringement is a complex legal issue and outside the scope of this paper.

**4.4 The DSA does not provide for secret key distribution**

**Response:**

The DSA does not provide for secret key distribution because the DSA is not intended for secret key distribution. In many applications a digital signature capability for integrity and nonrepudiation is sufficient and secret key distribution is not necessary. NIST does recognize the need for secret key distribution in other applications (e.g., where encryption is used). However, NIST and NSA have not yet selected such a method. NIST decided that it would be better to provide a public key based signature system immediately than to wait for both a signature system and secret key distribution system at some later time.

In addition, there are certain advantages to having separate algorithms for signature and key distribution. First, cryptographic algorithms that do not encipher data clearly come under the Department of Commerce export rules whereas export of encryption algorithms is controlled by the Department of State procedures which tend to be more restrictive [NBUL]. Secondly, certain countries readily permit the use of signature algorithms within their borders, but they restrict the use of encryption algorithms.

**4.5 The DSA is incomplete because no hash algorithm is specified**

**Response:**

On January 30, 1992 [FRSHS], NIST proposed a Secure Hash Standard (SHS) [DFIPSYY] which specifies a Secure Hash Algorithm (SHA) that is required for use with the DSA and whenever a secure hash algorithm is needed for federal applications. Copies of the SHS may be obtained by writing to the Standards Processing Coordinator (ADP), National Institute of Standards and Technology, Technology Building, room B-64, Gaithersburg, MD 20899. The SHA produces a 160-bit message digest on any data string up to $2^{64}$-1 bits. The SHS comment period ended on April 30. Comments were received from twenty-four separate government agencies, companies, and private individuals. The vast majority of the comments were favorable, and no technical flaws in the algorithm were found. NIST now plans to proceed with the process of making the proposed SHS a FIPS.

Table 1 shows sample SHA processing rates obtained for C code implementations of the SHA on three different computers. Other implementors may obtain differing rates based upon the degree to which the code has been optimized, the compiler used, and other factors. The rates appear adequate for many data security applications.

| Machine | Rate (bytes/second) |
|---|---|
| AT | 2,523 |
| 486 (33 MHz) | 28,169 |
| SUN SPARC | 222,233 |

**Table 1:** *Sample SHA Processing Rates*

**4.6 The DSA is not compatible with IS 9796**

International Standard 9796 [IS9796] is a standard for digital signatures with message recovery. According to this standard the message must be half the block size of a reversible public key encryption algorithm. The message is then redundantly padded to fill the entire block size and then "encrypted" with the user's private key to form the signature. An n-bit message results in a 2n-bit signature. Any verifier of the signature can use the public key of the signer to recover the redundantly encoded message. Rather than having the signer send the message as well as the signature, IS 9796 permits the recovery of the message from the signature itself.

**Response:**

IS 9796 specifies a digital signature scheme which provides message recovery from the signature. It is inefficient for signing moderate or long messages one half block at a time. The standard does allow for signing a message digest instead of a message, but then one would have to transmit the message along with the signature and the reversibility of the algorithm would provide no apparent advantage.

Since the DSA is not reversible, it could not meet the requirements of IS 9796 for a reversible algorithm. However, producing a 2n-bit signature from an n-bit message (as with IS 9796) is inefficient and causes unnecessary data expansion. When the DSA is used with the SHA algorithm an n-bit message will result in a 320-bit signature, and only n+320 bits need be transmitted. Thus, messages longer than 320 bits, or shorter than the block size minus 320 bits, will have less data transmission requirements if signed using the DSA.

In addition, there have been proposals for an alternative international signature standard, called "Digital Signature with Appendix". This alternative standard would permit the use of nonreversible algorithms for digital signatures and would not require that a n-bit message produce a 2n-bit signature. NIST will propose that the DSA algorithm be one of the algorithms that may be used in conjunction with the proposed alternative standard.

**4.7 The modulus is fixed at 512 bits**

Some parties responding to the request for comments believed that the DSA was insecure because the modulus was fixed at 512 bits. Others felt that although 512 bits provided adequate security for most of today's applications, it was not adequate for public key certificates and long term security.

**Response:**

The security of the DSA is based on the difficulty of solving the discrete log problem. Most security experts consider the discrete log problem to be at least as difficult as factoring (i.e., solving $y = g^x \bmod p$ for x is as difficult as solving $n = a * b$ for a and b when p is the same size as n). Therefore, the 512-bit DSA is at least as secure as many products, whose security is based on factoring, that are currently on the market today. One responder estimated that today it would take over eight million dollars (2.1 million MIPS[2]-years @ $4 per MIPS-year) to break the DSA but recommends allowing a modulus size of at least 710 bits.

Currently, smart card systems have limited computational capabilities which would be heavily utilized in implementing a 512-bit public key algorithm. Smart card implementations of larger modulus sizes are not yet practical. However, implementing a 512-bit algorithm in a smart card where the private key never needs to leave the card may offer much greater overall security than implementing a larger size modulus in a shared PC.

In response to the comments that a larger modulus size is required for certificates and long term security, modulus sizes of up to 1024 bits will be allowed. The revised standard will allow modulus sizes of 512, 576, 640, 704, 768, 832, 896, 960, and 1024 bits. This array of sizes should be sufficient for protecting sensitive unclassified data for the foreseeable future.

**4.8 The 160-bit size of q is too small**

**Response:**

Some parties claimed that the 160-bit size of q is too small but no analytical justification for this claim was provided. The 160-bit q provides a work factor of $2^{80}$ which is consistent with the 160-bit message digest provided by the SHA. (Note that the 160-bit SHA message digest is already 32 bits longer than most other accepted message

---

[2] MIPS = Million Instructions Per Second.

digests.) Assuming $32 \times 10^{12}$ operations per MIPS-year and a cost of $4 per MIPS-year, one would expect to spend at least [($2^{80}$ operations)/($32 \times 10^{12}$ operations/MIPS-year)] x ($4/MIPS-year) = $151,000,000,000 to recover a single key, x. It has been estimated by Andrew Odlyzko that this is roughly the same effort that would be required to break a discrete log system with a 1024-bit modulus using the number field sieve. Therefore, the 160-bit q appears to be sufficient even when a 1024-bit p is used.

**4.9 Compromise of k would compromise the private key**

**Response:**

Compromise of k would compromise the private key x. However it has not been shown that compromising k is any easier than compromising x itself. Both x and k are randomly or pseudorandomly generated; both x and k are kept in the most secure area of the cryptographic module; and neither x nor k need be known to any human being. If an adversary can gain physical access to k, then the adversary could also gain physical access to x. The DSA is designed so that neither x nor k can be determined from the signature.

NIST will suggest techniques for generating the x, k, and other values in an appendix of the DSS. In addition, the authors highly recommend the use of smart cards to protect private keys and any other secret parameters used by public key algorithms.

**4.10 Weak values of p could be selected by a dishonest CA**

A claim was made that a dishonest Certification Authority (CA) could purposely select a value of p for its own users which would permit the CA to recover the private keys of the users.

**Response:**

The proposed DSS specifies a Digital Signature Algorithm. It does not discuss all the ways the algorithm may be used or misused. The qualifications section of the DSS Announcement states that "The responsible authority in each agency or department shall assure that an overall implementation provides an acceptable level of security." The proposed DSS specifically states that, "Systems for certifying credentials and distributing certificates are beyond the scope of this standard." Therefore, one would not expect an algorithm specification standard to cover the case of a dishonest certification authority.

The DSS allows users to generate their own primes, p and q. The DSS also allows the user to use primes generated by a trusted party or a certification authority. If primes are known to be randomly generated, the user can even accept primes generated by a distrusted party. One can construct special primes that are considered weak. If they were used the private keys of the users might be recovered. (Note that many other algorithms have similar weak values.) However, the probability of generating a weak prime at random is infinitesimally small. (The probability of generating a weak p at random has been estimated to be less than $10^{-90}$.) Two parties pointed out that the use of a one-way function, such as the SHA, in the process that generates p and q could ensure that weak values occur only randomly. By making publicly known the input to the SHA, the resulting p, the resulting q, and the process, the user would be able to verify that weak primes were not purposely constructed. A technique which makes use of the SHA in the generation of DSA primes is proposed in Appendix A of this paper.

The claim that a trapdoor was purposely placed in the DSA was the subject of a panel session at Eurocrypt '92. No evidence of an intent to put a trapdoor in the DSA was presented and by the end of the session the claim was substantially discredited.

Warning! As with all systems using a certification authority, the certification authority must be trusted to correctly establish the binding between the user's identity and the user's public key.

**4.11 The DSA is less efficient for verification**

**Response:**

Some of the comments provided inaccurate estimates of the computation time required for the DSA. Obviously one would like a signature algorithm to be as efficient as possible while still providing adequate security. The real issue is whether the DSA verification speed is sufficient. On a 386 personal computer[3], the DSA can validate a signature in less than one second and the same computation can be done in milliseconds in hardware. These times are adequate for most applications.

In order to fully understand the computational differences between the DSA and RSA one must consider five different computations: global computations, key generation computations, pre-computations, signature computations and verification computations.

> *Global computations* may be performed once for a set of users and need not be recomputed for a long period of time. Therefore, these computations do not normally impose a severe penalty on the operational system. For the DSA, the computation of p, q, and g could be considered global computations. The RSA does not have a similar computation.
>
> *Key generation computations* are performed in generating the public and private keys. For DSA one must generate x and y as the private and public keys. For RSA, primes p and q must be generated and e and d computed. (Note that when using the Chinese remainder theorem, d mod (p-1) and d mod (q-1) are generated instead of d.)
>
> The *pre-computations* for the DSA are performed for each message to be signed. However, these computations may be performed before any message is selected to be signed. These pre-computations involve generating $k^{-1}$ and r as inputs to the signature generation computation. RSA has no similar computation.
>
> For DSA the *signature generation computations* involve generation of the message digest, H(m), and the s portion of the signature. For RSA signature generation, one must compute $s = (H(m))^d \bmod n$.
>
> When performing the *signature verification computations* the DSA computes a putative r from the received message m, the received r, and the received s. If the computed value of r equals the received value of r the signature is verified. Otherwise the signature is rejected. Using the RSA one computes $s^e \bmod n$ and compares it to the message digest of the received message.

Table 2 indicates some sample computation times for the DSA and RSA algorithms performed either in a Hitachi H8-310 smart card processor or in a host personal computer. Efficient smart card implementations of public key cryptography are difficult to achieve because of the limited capabilities of current 8-bit smart card processors. On faster computers or special purpose smart card processors, the differences in computation times between the DSA and RSA algorithms become less significant to the human observer.

The DSS offers an advantage with regard to its extremely efficient computation of the private and public keys. The private key is any randomly generated 160-bit value called x and the public key is y where $y = g^x \bmod p$. Since both computations are efficient the private and public keys can be easily generated on a smart card. While the public key can be read from the smart card at any time, the private key never needs to leave the protection of the card. Observed DSA key generation computations are 40-80 times faster than RSA key generation computations.

---

[3] Products are mentioned in this paper for informational purposes only and do not constitute an endorsement.

In addition, the DSS has the capability of performing most of the signature computations before the actual message to be signed has been selected. This is done by pre-computing k, $k^{-1}$, and r. In fact several k, $k^{-1}$, and r values may be precomputed in a fashion that is transparent to the user. Then, when the user selects the message or data to be signed, the signature will be computed in a fraction of a second. This feature is especially useful in today's smart card systems where the card will perform the necessary pre-computations while the user is selecting and forming the message to be signed. Therefore, the signature process appears very efficient to the user.

| Algorithm | DSA | RSA | DSA Common p,q,g Estimated |
|---|---|---|---|
| Global Computation | Off Card (P) | NA | Off Card (P) |
| Key Generation | 14 | Off Card (S) | 4 |
| Pre-computation | 14 | NA | 4 |
| Signature | .03 | 15 | .03 |
| Verification | 16 On Card<br>1-5 Off Card (P) | 1.5 | 10 On Card<br>1-3 Off Card (P) |

**Table 2:** *Smart Card DSA & RSA Computation Times* (All times are given in seconds. Off card computations performed on a 386, 33 MHz, personal computer. (P) indicates public parameters off card and (S) indicates secret parameters off card. Both algorithms use a 512-bit modulus.)

The signature verification computations for the DSA require more computations than signature generation and 10-15 times more than for the RSA algorithm. However, verification involves only public keys and can therefore be implemented in personal computers or in some other medium where more computational capability exists. This is an important distinction in smart card systems where signature generation would be performed in the secure card having a modest computational capability while verification could be performed elsewhere.

The DSA parameters p, q, and g can be selected by individual users or be common to a group of users. If individually selected, they must be passed along with the user's public key to anyone desiring to verify that user's signature. If common values are selected by a group of users, they need not be transmitted with each message or each user's public key. Efficiency is improved by reducing the number of parameters that have to be transmitted and by permitting the one-time computation of certain intermediate results.

When common or preestablished public values are employed, a technique due to Brickell, Gordon, and McCurley [BRICKELL] can be used to reduce the DSA pre-computation and verify times. NIST estimates that the pre-computation time can be reduced to approximately 1/4 the un-optimized time and the verify time to 1/2 the un-optimized time in a smart card implementation. Computer programs which make use of this work are now being developed at NIST and Sandia Laboratories.

In summary, the DSA validation computation appears to be adequate for many government and commercial applications. The DSA generates keys very efficiently and provides a pre-computation feature that can make the signature computation transparent to the user. Verification, although less efficient, is adequate for nearly all applications. These features may make the DSA highly desirable for many applications involving smart cards.

### 4.12 The DSS is "buggy"

One responder claimed that the DSS is "buggy" because if s = 0 then the computation of $s^{-1}$ at signature verification would "blow up". In addition if s = 0, then the user's private key x could be recovered.

**Response:**

The computation of $s^{-1}$ would not "blow up" on the verification calculation because the standard clearly states that the signature is rejected for any received s' outside of the range $0 < s' < q$. As far as the security issue is concerned, it is true that if $s = 0$ then x could be recovered. However, there is no need to check for a condition which occurs with probability $2^{-160}$. The proposed DSS allows implementors to either check for $s = 0$ upon signature generation or to ignore the unlikely event depending on their own preferences.

## 5. Future Efforts

The following set of activities are presently planned by NIST in adopting a DSS as a FIPS:

1. Complete analysis and summary of comments;
2. Analyze and attempt to resolve patent issues;
3. Develop and evaluate alternative signature certification authority infrastructures;
4. Propose technical enhancements to DSA;
5. Issue second solicitation of comments on revised DSA;
6. Hold a symposium on the applications of the DSA;
7. Investigate the economic interests involving the DSS;
8. Coordinate and harmonize the revised DSS with ANSI and ISO standards activities;
9. Conduct final coordination of DSS within government;
10. Recommend Secretary of Commerce approval of DSS;
11. Publish revised DSS after Secretary of Commerce approval.

A brief discussion of some of the major activities are presented below.

### 5.1 Resolution of Patent Issues

NIST is presently attempting to resolve the patent issues in accordance with its desire to make the manufacture, sale and use of devices and systems implementing the DSA free of royalties for patents. The U.S. government already has rights to use patented techniques assigned to Public Key Partners because the government sponsored some of the research leading to the patents. However, private users presently do not enjoy such rights. Neither the U.S. government nor private users presently enjoy rights to the Schnorr patent. Alternative solutions to potential problems are being reviewed.

### 5.2 Second Federal Register Solicitation of Comments

As currently envisioned, the DSS will be revised to allow the use of a larger modulus, to add a new method for generating p and q, to add a method for pseudorandom generation of k values, and to correct or clarify minor editorial and technical issues. In order to assure an adequate opportunity for review of the revised proposed DSS, NIST is planning to publish the proposal for a second comment period.

### 5.3 Applications Symposium

NIST plans to host a Symposium on the Applications of Digital Signature Technology. The purpose of the symposium is to provide a forum for discussion of common problems, goals, and issues pertaining to the application of the DSA. Further information will be provided as plans develop.

### 5.4 International Infrastructure

NIST is studying the legal and technical issues related to development and operation of an international digital signature infrastructure. The infrastructure would be a system of organizations, people and computers used for distributing certificates to individuals, government agencies and private companies. The study will examine the legal and regulatory requirements which must be addressed, propose a certification authority architecture, and attempt to clarify the roles that various government agencies wish to perform. Several U.S. government agencies are participating in and financing the study.

NIST perceives a great need for such an infrastructure. Electronic filing of corporate and personal tax returns could be made more efficient and more secure if such a structure were available. Federal payments to contractors, vendors and social security recipients could be fully automated if the integrity and authenticity of electronic payments were assured. An international infrastructure is needed to provide security for worldwide business communications. NIST is presently working with the federal organizations responsible for such large scale applications. NIST intends to hold workshops with potential users and knowledgeable technical people in order to develop an infrastructure that will meet these anticipated needs.

It is intended that the infrastructure will utilize existing concepts and systems. International Standard X.509 (a security part of the Directory standard) describes a tree structure for certifying digital signatures. A digital signature certificate distribution system has been designed in conjunction with the Privacy Enhanced Mail project. NIST plans to build on these efforts to produce a recommendation for consideration by federal organizations planning to use digital signatures. Results of the present study are anticipated in the middle of 1993.

## 6. Conclusion

Several milestones have been met and several still need to be accomplished. NIST will continue the work required for adoption of the proposed Digital Signature Standard as an approved Federal Information Processing Standard. NIST also believes that an international infrastructure is required in order for digital signatures to be widely used throughout the U.S. government and the world.

## References

[BRICKELL] E. Brickell, D. M. Gordon, K. S. McCurley, D. Wilson, Fast Exponentiation with Precomputation, Eurocrypt '92 Extended Abstracts, p 193-201.

[CSA87] Computer Security Act of 1987, June 11, 1987, Sec. 3.

[DANSIX9] Working Draft American National Standard X9.30-199X, Public Key Cryptography Using Irreversible Algorithms for the Financial Services Industry: Part 1: The Digital Signature Algorithm (DSA), American Bankers Association, Washington, DC.

[DFIPSXX] Draft Federal Information Processing Standards Publication XX, Announcement and Specifications for a Digital Signature Standard (DSS), August 19, 1991.

[DFIPSYY] Draft Federal Information Processing Standards Publication YY, Announcement and Specifications for a Secure Hash Standard (SHS), January 22, 1992.

[FRDSS] A Proposed Federal Information Processing Standard for Digital Signature Standard (DSS), Federal Register Announcement, August 30, 1991, p 42980-41982.

[FRSHS] A Proposed Federal Information Processing Standard for Secure Hash Standard, Federal Register Announcement, January 31, 1992, p 3747-3749.

[GAO91] Comptroller General of the United States Decision, Matter of: National Institute of Standards and Technology--Use of Electronic Data Interchange Technology to Create Valid Obligations, file B-245714, December 13, 1991.

[IS9796] Information technology - Security techniques - Digital signature scheme giving message recovery, IS 9796, International Organization for Standardization, Geneva, Switzerland.

[NBUL] NCSL Bulletin, Data Encryption Standard, Exportability of DES Devices and Software Products, June 1990, p 3-4.

[NLET] Letter to General Counsel, U.S. General Accounting Office, from James H. Burrows, Director of NIST Computer Systems Laboratory, September 13, 1990.

[RIVEST] R. Rivest, A. Shamir, and L. Adleman, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Communications of the ACM, No. 2, p 120-126, 1978.

## Appendix A: Generation of Primes p and q

The Digital Signature Standard requires two primes, p and q, satisfying the following three conditions:

a) $2^{159} < q < 2^{160}$

b) $2^{L-1} < p < 2^{L}$ for a specified L, where $L = 512 + 64j$ for some $0 \le j \le 8$

c) q divides p-1.

This prime generation scheme starts by using the SHA and a user supplied SEED to construct a prime, q, in the range $2^{159} < q < 2^{160}$. Once this is accomplished, the same SEED value is used to construct an X in the range $2^{L-1} < X < 2^{L}$. The prime, p, is then formed by rounding X to a number congruent to 1 mod 2q as described below.

An integer x in the range $0 \le x < 2^{g}$ may be converted to a g-long sequence of bits by using its binary expansion as shown below:

$$x = x_1*2^{g-1} + x_2*2^{g-2} + \ldots + x_{g-1}*2 + x_g \rightarrow \{ x_1,\ldots,x_g \}.$$

Conversely, a g-long sequence of bits $\{ x_1,\ldots,x_g \}$ is converted to an integer by the rule

$$\{ x_1,\ldots,x_g \} \rightarrow x_1*2^{g-1} + x_2*2^{g-2} + \ldots + x_{g-1}*2 + x_g.$$

Note that the first bit of a sequence corresponds to the most significant bit of the corresponding integer and the last bit to the least significant bit.

Let L - 1 = n*160 + b, where both b and n are integers and $0 \le b < 160$.

Step 1. Choose an arbitrary sequence of at least 160 bits and call it SEED. Let g be the length of SEED in bits.

Step 2. Compute

$$U = SHA[SEED] \text{ XOR } SHA[(SEED+1) \bmod 2^g].$$

Step 3. Form q from U by setting the most significant bit (the $2^{159}$ bit) and the least significant bit to 1. In terms of boolean operations, q = U OR $2^{159}$ OR 1. Note that $2^{159} < q < 2^{160}$.

Step 4. Use a robust primality testing algorithm to test whether q is prime[1].

Step 5. If q is not prime, go to step 1.

Step 6. Let counter = 0 and offset = 2.

Step 7. For k = 0,...,n let

$$V_k = SHA[(SEED + offset + k) \bmod 2^g].$$

Step 8. Let W be the integer

$$W = V_0 + V_1 * 2^{160} + \ldots + V_{n-1} * 2^{(n-1)*160} + (V_n \bmod 2^b) * 2^{n*160}$$

and let $X = W + 2^{L-1}$. Note that $0 \le W < 2^{L-1}$ and hence $2^{L-1} \le X < 2^L$.

Step 9. Let c = X mod 2q and set p = X - (c-1). Note that p is congruent to 1 mod 2q.

Step 10. If $p < 2^{L-1}$, then go to step 13.

Step 11. Perform a robust primality test on p.

Step 12. If p passes the test performed in step 11, go to step 15.

Step 13. Let counter = counter+1 and offset = offset + n + 1.

Step 14. If counter $\ge 2^{12} = 4096$ go to step 1, otherwise (i.e., if counter < 4096) go to step 7.

Step 15. Save the value of SEED and the value of counter for use in certifying the proper generation of p and q.

---

[1] A robust primality test is one where the probability of a non-prime number passing the test is at most $2^{-80}$.