An Educational System in Public Key Cryptography

Irina Noninska

Abstract: *The paper deals with mathematics and main steps for key pair generation in two asymmetric cryptographic algorithms – RSA and DSA. An educational system for implementation of digital signatures with appendix for electronic document authentication is proposed. The main goal of the system is to give students theoretical and practical knowledge in public key cryptosystems. It could be useful for e-learning in cryptography and e-business.*

Key words: *Public Key Cryptography, Cryptographic Protocols, Educational System, E-document Authentication.*

1. INTRODUCTION

Nowadays many people exchange personal and business information via the Internet. They would like to be sure that electronic documents, which they send and receive have been successfully protected against unallowed access and modification. It is important to bear in mind also that all business activities require in addition strong authentication of the users. Authentication could guarantee the content of an electronic document (e-document) and prove its origin, as well. Asymmetric cryptographic algorithms have been widely applied last twenty years in different methods and schemes for digital signature generation. Many of them – Fiat-Shamir, Schnorr, ElGamal, Rabin and Merkle, were designed especially for this purpose. One of the most efficient asymmetric algorithms is RSA, which supports not only key pair generation, but encryption using public key and digital signature generation with message recovery. Another well known and wide spread asymmetric algorithm – DSA, which is designed only for the purposes of digital signature generation, was proposed in DSS [1]. It implements a scheme, based on hashing algorithm – SHA-1, where the result is digital signature with appendix and the reverse procedure – verification requires the original e-document as input. Recently public key standards recommend implementation of hash algorithms- MD5 and SHA-1 in RSA-signature systems in order to prove data integrity by hashing first and then sign the hash value using sender's secret key to generate digital signature. As a result all digital signature schemes that use one-way hash algorithm could be considered as reasonably robust and stable to cryptanalytic attacks. On the base of DSA and elliptic curve theory, Elliptic Curve Digital Signature Algorithm (ECDSA) was proposed. These three algorithms, mentioned above – RSA, DSA and ECDSA have been recommended by Bulgarian low of electronic document and digital signature, which came into force in 2001.

Asymmetric cryptographic algorithm could be applied not only for digital signatures generation, but in different authentication schemes, based on the principles of checking what the user is allowed to propose, proving that he is the right person. For example, it could be something he knows – passwords, PIN-codes, secret parameters for handshaking, etc. or something he has obtained to take part in a session – access ticket, secret cryptographic key, which is used to generate digital signature or just a parameter for one-way recognition. All methods, algorithms, standards and protocols, used for strong authentication, data integrity and no repudiation by digital signatures on the base of key pair (public/secret) define the subject of public key cryptography [3,4].

The paper discusses basic valuable security services that public key cryptosystems provide – digital signature generation and verification, authentication protocols, notarization of e-document. An educational system, which presents different protocols and methods applicable in e-business operations, is proposed. The main goal is to give students theoretical and practical knowledge in public key cryptosystems, their implementation in electronic document processing and authentication schemes.

2. RSA-ALGORITHM – KEY GENERATION AND ENCRYPTION

Three researchers – Rivest, Shamir and Adleman first published details of the RSA-algorithm in 1978 [2] as a result of their efforts and towards minimizing key complexity of ciphers and their number. For example, if a symmetric cryptosystems has "*n*" users, the necessity keys are on the order of "n^2" in comparison with "2*n*", required in asymmetric cryptosystem, proposed by the authors.

RSA-algorithm could be explained by 6 steps as shown in fig. 1 where steps 1-4 present RSA-mathematics, step 5 – encryption of a plain text *P* and step 6 – decryption of the cryptogram *C*.

Step (1): The user *A* chooses two large primes – *p* and *q*. They are secret parameters.

Step (2): *A* multiplies *p* and *q*. The result is N : $N = p.q$.

Step (3): *A* calculates the Oilier function $\Phi(N)$: $\Phi(N) = (p-1).(q-1)$.

Step (4): *A* generates his own secret key $SK_A : 0 < SK_A < \Phi(N)$ and SK_A relatively prime to $\Phi(N)$.

A calculates his own public key PK_A as multiplicative inversion, i.e. $PK_A = SK_A^{-1} \bmod \Phi(N)$.

The problem of modular inverse sometimes has a solution, sometimes not. In general it might have right solution only when $\Phi(N)$ and SK are relatively prime, otherwise there is no solution. When $\Phi(N)$ is a prime, there is the only one inverse value in that range.

Step (5): The user *A* shares his public key PK_A and *N* with the user *B*. If the plain text is *P*, the user *B* can encrypt it as follows: $P^{PK_A} \bmod N = C$.

Step (6): The user *A* receives the cryptogram *C* from the user *B*. He is the only person who can calculate the plain text *P* from *C*, because only he knows the key – his own secret key SK_A. He calculates *P* as follows: $P = C^{SK_A} \bmod N$.

Fig.1 RSA-algorithm – Key Generation

3. DIGITAL SIGNATURES ENABLE E-DOCUMENT AUTHENTICATION

Digital signature schemes could be divided into two groups depending on requirements of the secret and public parameters and complexity of two procedures – generation and verification.

The first group includes systems based on digital signature with appendix, where an e-document should be cryptographically processed by the secret key of the sender after hashing. At the same time digital signature of one and the same e-document changes, because except hash function, generation process uses an additional secret parameter with different value when this document must be signed several times. Typical representatives of this group are algorithms ElGamal, Schnorr and DSA.

The second type of digital signature systems is based on message recovery. This mechanism is used by algorithms of Rabin, Nyberg-Rueppel and RSA. They allow verifying digital signature using only obtained signature and the public key of the sender. When a hash algorithm is used – SHA-1 for example, it is able to process plain text of 2^{64} bits to hash code of 160 bits. This is the reason to apply RSA with hash algorithm ensuring short hash codes of e-documents. As a result representatives of this group could propose invulnerable to attacks digital signature schemes.

Fig. 2 presents main steps of DSA, digital signature generation and verification.

Step (1): User *A* chooses two large primes numbers $2^{L-1} < p < 2^{L}$, where $512 \le L \le 1024$; $2^{159} < q < 2^{160}$; *q* divides evenly (*p-1)*.

Step (2): User *A* chooses a number *h*, smaller than (*p*-1) and computes $g = h^{\frac{p-1}{q}} \bmod p$.

Step (3): User *A* chooses a random number, which is his secret key $SK_A : 0 < SK_A < q$.

Step (4): User *A* calculates his public key $PK_A : PK_A = g^{SK_A} \bmod p$.

Digital signature generation

User *A* signs the plaintext *P*.

Step (1): User *A* generates a random secret parameter *k* which has different values for each signature: $0 < k < q$.

Step (2): User *A* calculates two values – *r* and *s*, which constitute digital signature of *P*. He calculates also hash value of *P* : $h(P)$:

$$r = (g^k \bmod p) \bmod q;$$

$$s = (k^{-1}(h(P) + r.SK_A)) \bmod q.$$

User *A* sends his message *P* to user *B* with digital signature as appendix, as follows: (P, r, s)

Digital signature verification

Step (3): User *B* verifies the signature. For this purpose he calculates four parameters:

$$u = s^{-1} \bmod q;$$

$$v = (h(P).u) \bmod q;$$

$$w = (r.u) \bmod q;$$

$$z = ((g^v.PK_A^w) \bmod p) \bmod q.$$

Step (4): Finally, user *B* should check if *z* is equal to *r*. When *z=r*, the signature has been successfully verified, otherwise it must be rejected.

Fig.2 Digital Signature Generation and Verification Using DSA

Having in mind common security requirements of business activities via the Internet and necessity of e-document authentication, two cryptographic protocols on the base of RSA are proposed. They could be easily understand and implemented, hence they are useful for public key cryptography and e-business learning. The first one is a protocol for group signature. It is realized over four different groups of business partners who exchange e-documents, where maximal value of the users is defined *(n)* and every user has several keys (*m*). These keys are generated using RSA-algorithm. As a result 4*nm* pairs of cryptographic keys (PK/SK) are defined. Every member of a group can sign current e-document, choosing one of his own secret keys and get a group signature, which becomes a part of this e-document. After that he is allowed to disseminate signed e-document between all users of the system or to send it just to one of them. Every user is able to verify the group signature, but he could not reveal the real sender of the electronic document. It is possible to prove only that this e-document has been prepared from one the groups.

The second protocol is intended to sign e-documents with blind signature. It deals with a method, proposed by D. Chaum [5] and uses all cryptographic keys, generated for the group signature. In addition it involves as an authority *Notary*, who is responsible for signing the e-documents using his own secret key. The Notary has no rights to access the text of the e-document. He is allowed to sign each one, proving that this e-document has been sent him before sending to other users, but at the same time he is

not able to read the information in it. In this way, applying the protocol for blind signature, the process of authentication could be coordinated by the Notary, because he is the only person, who can prove the signature of each e-document.

4. PRACTICAL KNOWLEDGE IN PUBLIC KEY ALGORITHMS

For the purposes of education in public key cryptography a learning system is designed. It deals with two asymmetric cryptographic algorithms – RSA and DSA, presenting their mathematics and main steps for key pair signature. RSA is used to explain group and blind signature. The first task is to generate primes. Every student can choose how to get two primes – using a module of the educational system designed on the base of Rabin-Miller algorithm or entering his own code. After successfully prime generation, the rest parameters N and $\Phi(N)$ must be calculated. The second task is key pair generation. Finally all parameters – public and secret, should be checked by the RSA-control module of the system. If they are correct the student will receive message that confirms successfully done RSA scheme, otherwise he should try to reveal mistakes and propose new values, for *p* and *q* and calculates SK and PK. The system proposes RSA-module for key generation which could be used by users who have no good skills in programming. They can access, as well, samples with six steps of RSA-algorithm and their results, as shown in fig. 3.

$p = 1733$	$p = 17$
$q = 2347$	$q = 23$
$N = p.q = 1733 .2347 = 4067351$	$N = 391$
$\Phi(N) = (p-1).(q-1) = 1732.2346 = 4063272$	$\Phi(N) = 352$
$PK = 31$	$PK = 205$
$SK = 3145759$	$SK = 421$
$E_{PK}(P) = P^{31} \bmod 4067351 = C$	$E_{PK}(P) = P^{205} \bmod 391 = C$
$D_{SK}(C) = C^{3145759} \bmod 4067351 = P$	$D_{SK}(C) = C^{421} \bmod 391 = P$

Fig.3 Samples of RSA

RSA has been put at the root of many cryptographic protocols for authentication. Each security protocol consists of a series of steps that should be carried out from a person to prove that he is a trusted user of the system and all these steps should be designed according to preliminary defined security rules. The second module of the educational system presents two protocols for e-document authentication. They are designed on the base of RSA – algorithm using hash algorithms MD5 and SHA-1. The purpose is to enhance practical knowledge of students in public key cryptosystems. The first protocol is proposed for *group signature*. Four groups are defined. Each of them has three users. For the purposes of their communication 36 cryptographic key pairs are generated, hence every number of a group possesses three key pairs. He is able to generate digital signature, choosing one of his three secret keys randomly. Verification of the signature requires implementing of the corresponding public key from the list. *Blind signature* protocol realizes the idea of trust component in the system, which is responsible for authentication efficiency. This component – *Notary* must sign each e-document, confirming that is has been sent to him. For this purpose he possesses a key pair, which public key is included in the list of keys. The Notary uses the secret key of this pair to generate blind signature. His signature can prove the authentity of an e-document and guarantee required level of anonymity, because the content of the document is preliminary hidden by his sender. The Notary receives hash of e-document and since MD5 and SHA-1 are one-way functions, there is no doubt that he can not reveal the plain text. This protocol could be used in e-business applications, where

users insist on protecting their personal and business information from tampering and confirm the authentity of every e-document at the same time. Contemporary technologies for timestamping employ notarization for indicating the time when it was performed.

Fig. 4 presents functionality of the educational system. Every student can access two main modules – RSA and DSA. He can use text files, proposed by the system as e-documents to sign each one employing group or blind signature. Implementation of two hashing algorithms MD5 and SHA-1 allow to display differences between digital signatures of one e-document, obtained after signing its different hash values.

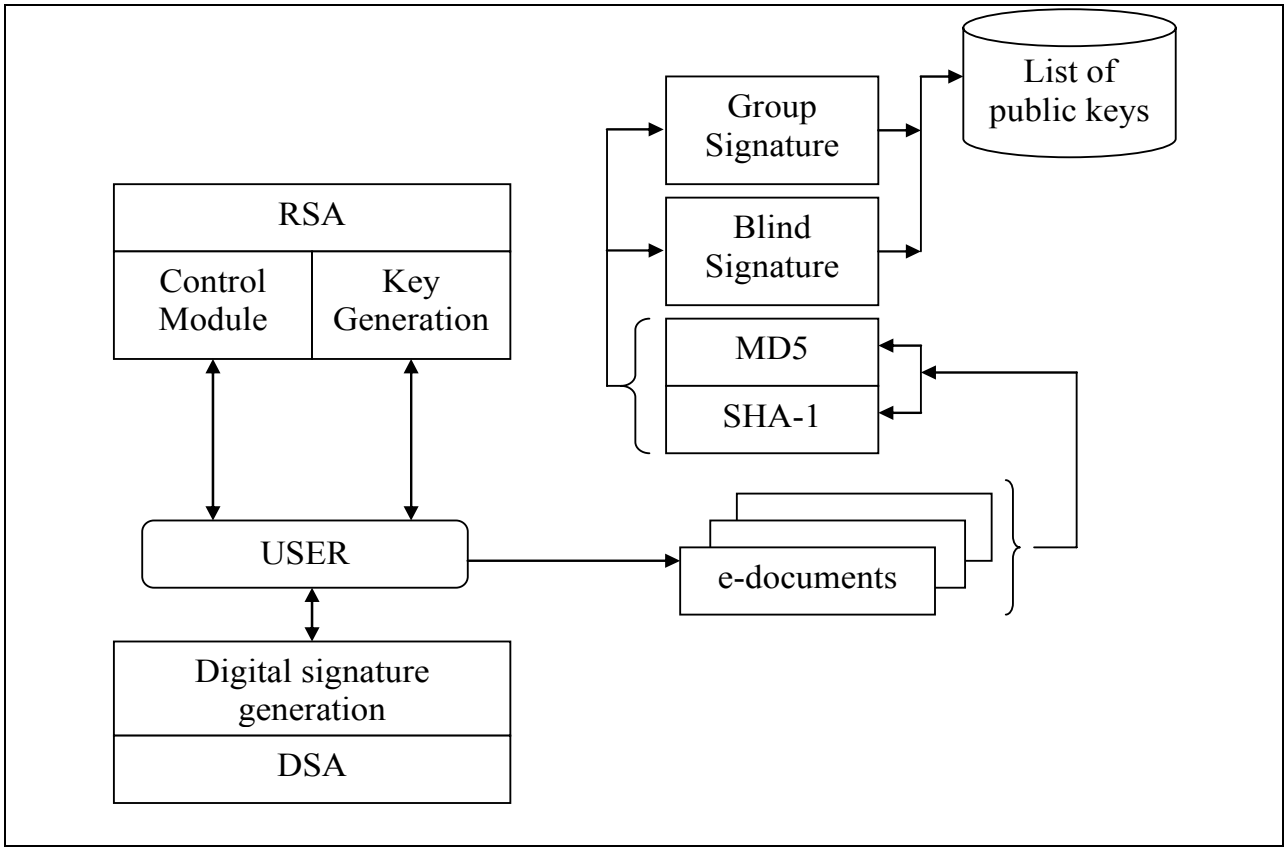

Fig.4 Educational System in Public Key Cryptography

Having in mind programming skills of the students and their common knowledge in Web-technologies main learning modules of the system are proposed using Microsoft Visual Studio 2008, C++ program language. Students can access three panels, designed to present main cryptographic schemes, described above. The first panel deals with implementation of RSA-algorithm for key pair generation. As shown in fig. 4, it is closely connected with group and blind signature modules, which could be applied after defining the cryptographic keys. The second panel is proposed for digital signature generation using DSA. In order to learn more about hashing procedures applied in public key cryptography, students could choose the last panel. It proposes results – hash values of e-documents obtained by MD5 or SHA-1.

5. CONCLUSION

The educational system, proposed in this paper is applicable for e-learning in cryptography and e-business. Students have opportunity to understand benefits that public key cryptography could provide to business. They learn more about principles of e-documents authentication. The system could be useful for users who would like to have more knowledge about design and implementation of asymmetric cryptographic algorithms in digital signature protocols.

REFERENCES

[1] The Digital Signature Standard (DSS). Communications of the ACM, v.35, 1992, 7

[2] Rivest, R., A. Shamir, A. Adleman, A Method for Obtaining Digital Signatures and Public Key Cryptosystems. Communication of the ACM, v.21, 1978

[3] RSA Technology Web Site (http://www.rsa.com)

[4] Stallings, W. Network Security Essentials: Applications and Standards, Prentice-Hall, 2000.

[5] Chaum, D. Blind Signature Systems, 1998.

[6] Secure Hash Standard, NIST, 1995.

[7] Mukesh, K. Public Key Cryptography With Matrices. Proceedings of the Fifth Annual IEEE SMC, 2004.

ABOUT THE AUTHOR

Assoc. Prof. Irina Noninska, PhD, Department of Computer Systems, Technical University – Sofia, Bulgaria, Phone: +359 2 965 34 71, E-mail: irno@tu-sofia.bg.