ELECTRONIC POWER OF ATTORNEY PROTOCOL BY USING DIGITAL SIGNATURE ALGORITHM

[1,3]WALIDATUSH SHOLIHAH, [2]SUGI GURITMAN, [3]HERU SUKOCO
[1]Diploma Programme, Bogor Agricultural University
[2]Mathematics Department, Bogor Agricultural University
[3]Computer Science Department, Bogor Agricultural University
E-mail: [1]walidah@ipb.ac.id , sugigu@ipb.ac.id , [3]hsrkom@ipb.ac.id

ABSTRACT

In this paper, we focused on making electronic power of attorney protocol. Power of attorney is a letter that authorizes the holder to carry out specified power given by the grantor. Generally power of attorney consist of the giver and holder identity, the contents of the letter and the signatures of the parties. The signature was made by using a tool such as pen. This power of attorney is certainly less efficient in time, resource, and security as well. Conventional signature (made with pen or other stationery) was easily faked and still using the naked eye for verification. These weaknesses can be overcome with the electronic power of attorney. Electronic power of attorney is using a digital signature for the giver and the holder. The signature used is Digital Signature Algorithm (DSA). Power of attorney in paper media can be replaced with electronic power of attorney. Signature on a power of attorney in paper media was replaced with a digital signature. Electronic power of attorney created using DSA algorithm according to the prevailing power of attorney in Indonesia. Electronic power of attorney protocols are designed to meet the security criteria of a power of attorney.

Keywords: *Digital Signature Algorithm, Electronic, Power of Attorney, Protocol, Signature*

1. INTRODUCTION

Power of attorney is a letter that authorizes the holder to carry power as specified in authorization issued by the authorizer. This power of attorney may be evidence for the holder of the letter. Power of attorney arrangement consists of the identity of the parties, the things that are authorized and signatures of the parties. Signature on a power of attorney created by using a tool such as ballpoin or pen. Both parties must be met to sign the letter. This kind of power of attorney is certainly less efficient in terms of time and security as well. Conventional signature (made with pen or other stationery) is easily faked and still use the naked eye for verification. These weaknesses can be overcome with the electronic power of attorney. Electronic power of attorney is using a digital signature to the holder and authorizer .

Research on electronic power of attorney has not been done. At The Peninsula Daily (http:// thepeninsulaqatar.com/qatar/224250-no-more-power-of-attorney-in-paper-format.html accessed on June 3, 2013), the law ministry of Qatar, want to replace the conventional power of attorney made on paper format into an electronic format that is more secure and efficient in terms of time. But it remains unclear whether the electronic power of attorney has been applied or not. In the UK, there is a website that serves electronic power of attorney (http://www.publicguardian-scotland.gov.uk/epoar / index.asp accessed on June 3, 2013). But only the delivery is being done online. Power of attorney remains in paper and then scanned. Although Indonesia has set the use of digital signature but have not been applied it to the electronic power of attorney.

Rules regarding digital signature contained in the Act of electronic information and technology No. 11 of 2008 chapter 1 verse 12. The digital signature technology is widely used in biometrics (4%), the data transaction (7.2%), e-messaging (10.4%), wireless security (5.6%), data access (10.4%), and other technologies (62.4%) [1].

Digital Signature Algorithm (DSA) is based on NP-complete problems such as prime factorization, discrete logarithm problem, elliptic curve problem, and others. DSA and ECDSA (Elliptic Curve DSA) algebraically equal [2]. Research on DSA has been done by researchers, such as: A Digital Signature Algorithm based on X^{th} Root Problem introduce hard number theoretical problem called X^{th} root problem [3]. X^{th} root problem is quite competitive compared with other DSA is based on multiple hard problems. DSA can be made by combining two hard problems (prime factorization and discrete logarithm problem) [4]. By combining these two hard problem, cryptanalis must resolve both problems at the same time to solve the algorithm. DSA scheme based on block cipher algorithm that examined more efficient and more secure [5].

DSA uses hash functions in the algorithm. Research on hash function has been done such as: a new hash algorithm called SHA–192 is more secure than SHA-1 [6] and the security level of hash function increase if the digest depends on the content of the message [7]. Method in [7] maybe more secure but it has a high complexity.

Proxy signature is a special form of digital signature. It was first proposed by Mambo, Usuda, and Okamoto in 1996 in a study entitled Proxy Signature: Delegation of the Power to Sign Messages. Proxy signature enables the original signer (the original signer) delegates his right to sign a document to another person (proxy signer) [8]. Proxy signature scheme can be used as electronic power of attorney. But the results of the proxy signature is only one signature. This scheme is not fit the conditions in Indonesia. The power of attorney in Indonesia needs two signatures.

From the above description it can be concluded that the power of attorney can be made in electronic form by using digital signatures. A power of attorney can be seen and read by anyone. But the contents of the original power of attorney shall remain (unchanged). Signature of giver and receiver using DSA and must be convinced belongs to the person concerned. Reasons for selecting DSA because the computer is still not able to compute the factorization of large prime numbers. DSA thus safe to use. On the electronic power of attorney , the signature is created using digital signatures with appendix so that the original file is not changed after the letter was signed.

This study has the objective to create a power of attorney electronic protocol using the Digital Signature Algorithm (DSA). The scope of this research are: 1) Protocol electronic power of attorney is a power of attorney will be made under the hand, not stamped, 2) Protocol not including protocol which made submission to the power of attorney form online and tax officials.

2. MATERIALS AND METHODS

The methodology of this research consists of several stages of process: analysis, the making of protocol, simulation and security analysis. The methodology of electronic power of attorney is shown in Figure 1.

2.1 Problem Analysis

At this stage, the problem is about creating a power of attorney that paperless. Power of attorney in paper media has many shortcomings and easily fabricated. The power of attorney may be used as evidence, so we need a way to get a power of attorney made in electronic form shall remain legal in front of the law.

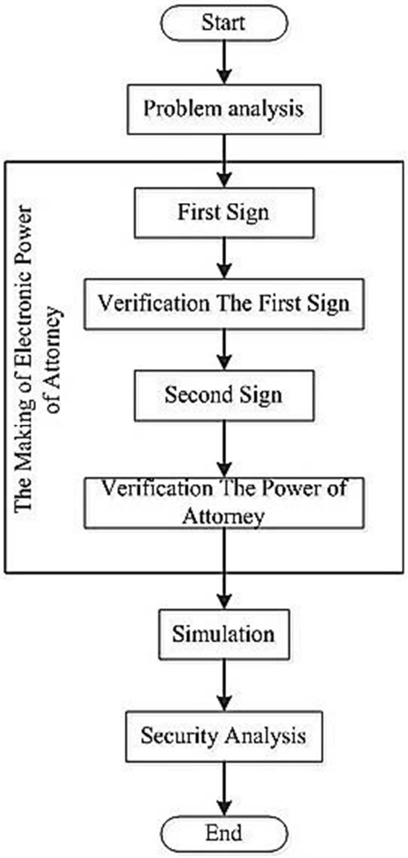

Start
Problem analysis
The Making of Electronic Power of Attorney
First Sign
Verification The First Sign
Second Sign
Verification The Power of Attorney
Simulation
Security Analysis
End

Figure 1: The Methodology Of Electronic Power Of Attorney

2.2 The Making of Protocol

The parties in this protocol are the authorizer (A), the holder (B) and third parties or verifier (C). The protocol diagram can be seen in Figure 2. The protocol of electronic power of attorney by using DSA is as follow:

1 A already has power attorney in a file. A generates his private and public key. Then A sign the file (S_A). B accept the file and S_A from A.
2 B verifies the power of attorney, which signed earlier, with A's public key. If it valid, then B generates his private and public key. Then B signed a power of attorney (S_B) with the corresponding private key. The power of attorney, S_A and S_B is sent to C for verification. Conversely, if verification by B is invalid, then the protocol fails.
3 C receives the power of attorney that has been signed by A and B. C check the power of attorney using the public key of A and B. If the results of the verification was appropriate, then the power of attorney was valid. If the result does not match the verification, power of attorney is not valid.

Signature of authorizer and holder using the Digital Signature Algorithm (DSA). Signing algorithm consists of three parts: key pair generation, signature generation and verification. DSA algorithm from Menezes (1997) is as follows:

1 Key pair generation
   a Select prime number q such that $2^{159} < q < 2^{160}$.
   b Choose t so that $0 \le t \le 8$, and select a prime number p where $2^{511+64t} < p < 2^{512+64t}$ with the property that q divides (p-1).
   c Select a generator α of the unique cyclic group of order q in $^{*}_{p}$
      (i) Select an element $g \in {}^{*}_{p}$ and compute $\alpha = g^{(p-1)/q} \bmod p$.
      (ii) If $\alpha = 1$ then go to step (i).
   d Select a random integer x so that $1 \le x \le q-1$
   e Compute $y = \alpha^x \bmod p$
   f Public key: (p, q, α, y), private key: x.
2 Signing
   a Select a random secret integer k, $0 < k < q$.
   b Compute $r = \left(\alpha^k \bmod p\right) \bmod q$.
   c Compute $k^{-1} \bmod q$.
   d Compute $s = k^{-1}\left\{h(m) + xr\right\} \bmod q$. $h(m)$ is hash function of message m.
   e Signature for m is the pair (r, s).
3 Verification
   a Obtain authentic public key (p, q, α, y)
   b Verify that $0 < r < q$ and $0 < s < q$; if not then reject the signature.
   c Compute $w = s^{-1} \bmod q$
   d Compute $u_1 = w.h(m) \bmod q$ and $u_2 = rw \bmod q$.
   e Compute $v = \left(\alpha^{u_1} y^{u_2} \bmod p\right) \bmod q$
   f Accept the signature if and only if $v = r$.

2.3 Simulation

This protocol simulated using Netbeans IDE 7.0.1 with the Java programming language. The process consists of four stages:

1 Making powers of attorney and signature of the authorizer (A)
   a Generating key pair (public key and private key) endorser. Public key is stored in a file.
   b Creating and initializing object signatures.
   c Sign the document.
   d Save the signature in a file
   e Input to this stage is a file message (m) . The output are signature and public key A (y_A).
2 Power of attorney verification by the holder (B). At this stage of the holder gets (m, S_A, y_A). It is the input to the verification algorithm. Output of this stage is "true" or "false".
3 The holder signature (B)
   The signing process by B is as same as A's signing process. The output of this process is public key B (y_B).
4 Power of attorney verification by verifier (C). The third party or verifier gets (m, S_A, S_B, y_A, y_B). That is input for verification algorithm. Output of this process is "true" or "false".

3. RESULTS AND DISCUSSIONS

The simulation of this protocol is using Netbeans IDE 7.0.1 with the Java programming language. There are three objects to be cast in this simulation. The first is the authorizer (A), the second is the holder (B) and the third party who will verify the letter (C). The first person fill a power of attorney form and then sign it. First appearance on the menu can be seen in Figure 3. In Figure 4, A make public key (pubA) and private key (signA). Public key and private key are encrypted so that when opened, it will show the characters that are not meaningful.

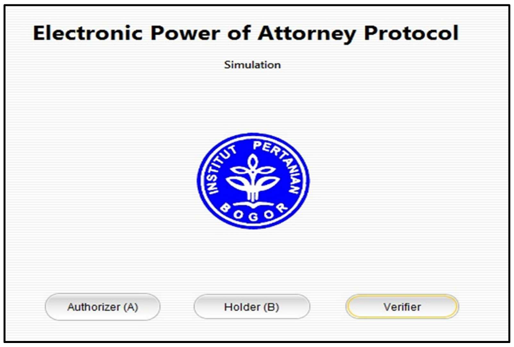
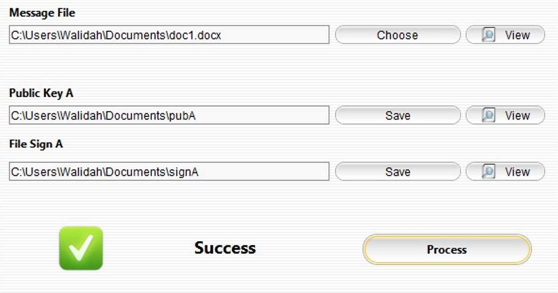


Figure 3: Protocol Simulation Application



Message File
C:\Users\Walidah\Documents\doc1.docx

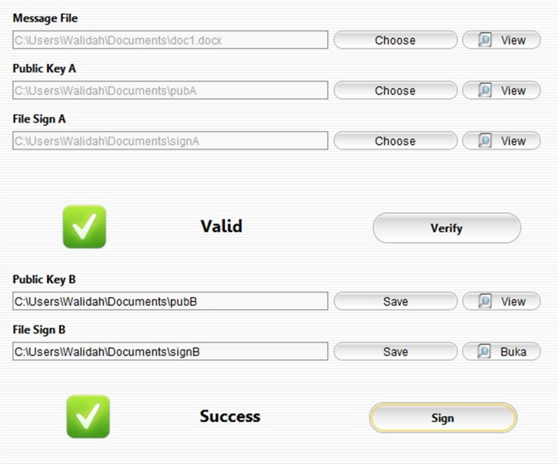
Public Key A
C:\Users\Walidah\Documents\pubA

File Sign A
C:\Users\Walidah\Documents\signA

Success

Figure 4: A Sign The Power of Attorney

A sign the letter with the corresponding private key. B receive letters from A. B then verify the power of attorney. If it’s valid, then B will sign it. If the power of attorney is not valid, then the menu for the second signature is not active (Figure 5). If this is the case then A must re-create the power of attorney or send back a letter that he had made to the second. Figure 6 shows the menu of third party verification. Figure 6 shows that a valid power of attorney. If there is one element that does not match then the app will bring up the words "not valid".



Message File
C:\Users\Walidah\Documents\doc1.docx

Public Key A
C:\Users\Walidah\Documents\pubA

File Sign A
C:\Users\Walidah\Documents\signA

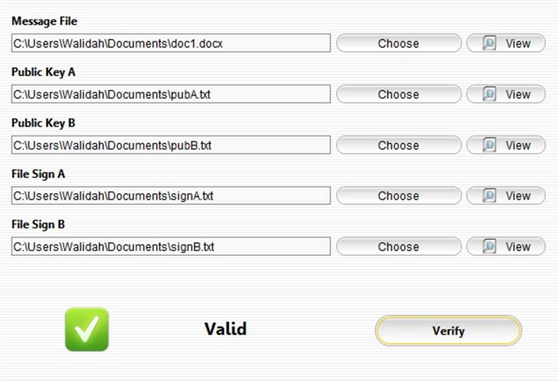
Valid

Public Key B
C:\Users\Walidah\Documents\pubB

File Sign B
C:\Users\Walidah\Documents\signB

Success

Figure 5: B Sign The Power of Attorney



Message File
C:\Users\Walidah\Documents\doc1.docx

Public Key A
C:\Users\Walidah\Documents\pubA.txt

Public Key B
C:\Users\Walidah\Documents\pubB.txt

File Sign A
C:\Users\Walidah\Documents\signA.txt

File Sign B
C:\Users\Walidah\Documents\signB.txt

Valid

Figure 6: Menu for Third Party (Verifier)

4. SECURITY ANALYSIS

DSA is secure because based on discrete logarithm problem. The discrete problem is still difficult to be solved by today's computers. In other words, the problem is said to be not viable count. The fastest algorithm to solve the problem today is the index calculus method. DSA security level will be higher if the selected primes greater than 1024 bits.

Electronic power of attorney, as a form of power of attorney must also meet the criteria of security. Based on the protocol of electronic power of attorney that has been created, it can be proven that the protocol electronic power of attorney has been made to meet safety criteria such as a power of attorney on paper media .

1 Everyone can read the electronic power of attorney. The most important part of the electronic power of attorney is the integrity of the letter.

2 The contents of the electronic power of attorney may not be altered. With the hash function (hash value of a power of attorney), a power of attorney may not be changed without changing the private key of the signer. Everyone who has public key can check if there is change in electronic power of attorney .

3 Signature on power of attorney must actually made by authorizer and holder. Signature authorizer and holder can be checked from the authorizer and holder's public key. If it matches, then of course the signature belongs to the authorizer and holder.

4 The authorizer and holder can not deny the signatures they have created. When they sign the electronic power of attorney, then all parties who have the authorizer and holder 's public key can easily check the signatures. Authorizer and holder certainly can not deny the signature they have created themselves.

The authorizer and holder generate public and private key together and that key are paired.

5 Signature authorizer and holder must be distinguished. Signature authorizer and holder can be distinguished from each private key. The authorizer and holder's private key generated by each of authorizer and holder without any interference from the other party .

6 The third party must be convinced that it is true authorizer give the authority to the holder. Electronic power of attorney signed two times, by the authorizer and holder. Third party verify the signature and the power of attorney through the public key authorizer and holder and also the message file itself.

Menezes (1997) said that the security of the DSA relies on two distinct but related discrete logarithm problems. One is the logarithmproblem in $_{p}^{*}$, where the powerful index-calculus methods apply; the other is the logarithm problem in the cyclic subgroup of order q, where the best current methods run in “square-root” time.

5. CONCLUSION

Power of attorney with paper media can be replaced with electronic power of attorney. Signature on a power of attorney in paper media was replaced with a digital signature. Protocols electronic power of attorney created using DSA algorithms according to a power of attorney in force in Indonesia. Electronic power of attorney protocols are designed to meet the security criteria of power of attorney.

REFERENCES:

[1] P. Shiralkar and B. S. Vijayaraman, “Digital Signature : Application Development Trends In E-Business,” *Journal of Electronic Commerce Research*, vol. 4, no. 3, pp. 94-101, 2003.

[2] J. Seberry, V. To, and D. Tonien, “A New Generic Digital Signature Algorithm,” *University of Wollongong Research Online*, vol. 3, pp. 221-237, 2011.

[3] K. K. Agrawal, R. Patira, and K. Madhur, “A Digital Signature Algorithm based on x th Root Problem,” *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 11, pp. 61-65, 2012.

[4] S. Vishnoi and V. Shrivastava, “A new Digital Signature Algorithm based on Factorization and Discrete Logarithm problem,” *International Journal of Computer Trends and Technology*, vol. 3, no. 4, pp. 653-657, 2012.

[5] P. Kuppuswamy, P. M. Appa, and S. Q. Y. Al-khalidi, “A New Efficient Digital Signature Scheme Algorithm based on Block cipher,” *IOSR Journal of Computer Engineering*, vol. 7, no. 1, pp. 47-52, 2012.

[6] T. Lakshmanan and M. Muthusamy, “A Novel Secure Hash Algorithm for Public Key Digital Signature Schemes,” *The International Arab Journal of Information Technology*, vol. 9, no. 3, pp. 262-267, 2012.

[7] M. Swarnkar and S. Verma, “Count based Secured Hash Algorithm .,” *IOSR Journal of Computer Engineering*, vol. 6, no. 6, pp. 49-51, 2012.

[8] S. Verma and B. K. Sharma, “A New Proxy Blind Signature Scheme Based on DLP,” *International Journal of Information and Network Security*, vol. 1, no. 2, pp. 60-66, 2012.

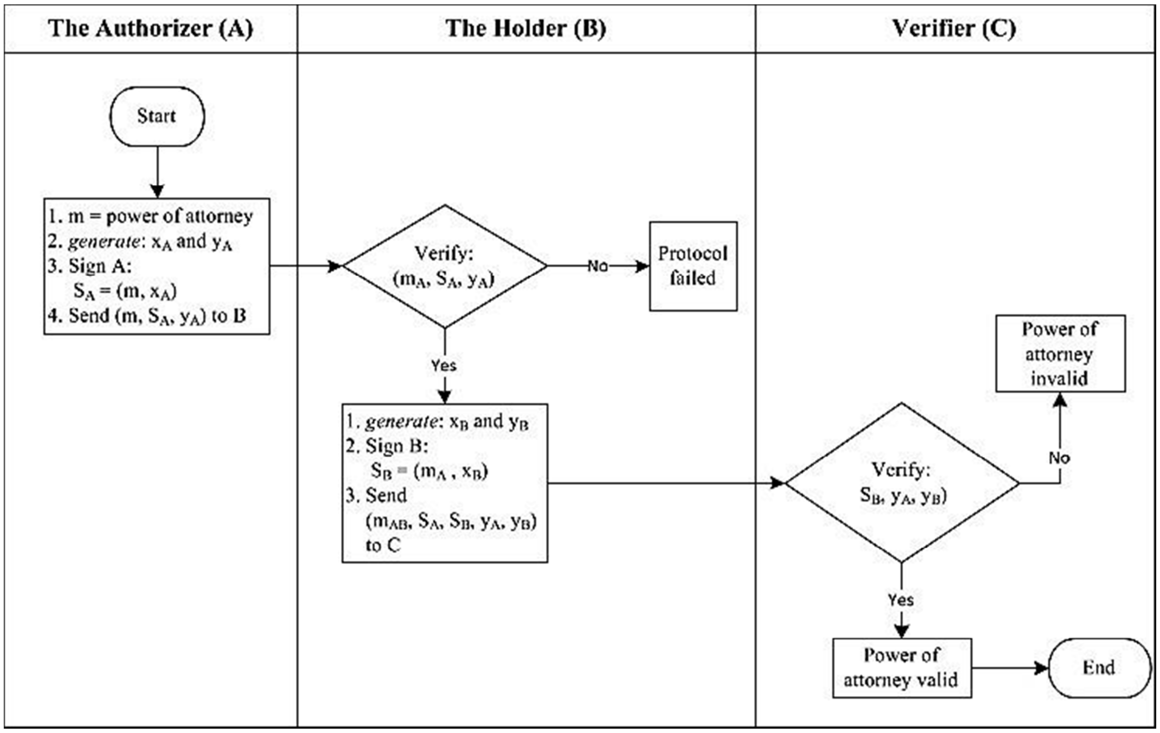


Figure 2: The Diagram of Electronic Power of Attorney