A Controlled Proxy-Protected Signature Scheme with Instantaneous Revocation

Khaled Shehata[†], Gamal Selim[††], Maged Elgindy[†††], Mohamed Mohamed Kouta [††††]
[†, ††]Arab Academy for Science and Technology
[†††, ††††]Egyptian Ministry of Finance

Abstract
We provide a proxy signature scheme using blind signature with instantaneous revocation. In the proposed schema, the original signer blinds his private key and sends it to delegated entity to sign on the message on behalf of him. The verifier in the proposed scheme will be able to verify the following upon receiving the message from delegated entity: 1) The message is coming from delegated entity (proxy signer), 2) The delegated entity is delegated from signer, 3) The Delegation Authority approves on this delegation, 4) The message is signed by the original signer. If the original signer wants to revoke this delegation he sends a revocation request to the delegation authority. The proposed schema inserts delegation authority to witness delegation like notary and verifies that the proxy signer does not violate the domain of delegation.

1. Introduction

Ordinary Signature Scheme is one of the most important primitive of public-key Cryptography. As a variation of ordinary digital signature scheme, a proxy signature allows a designated person, called a proxy signer (Delegated Entity), to sign the message on behalf of the original signer. Most proposed proxy signature schemes are based on discrete logarithm problems [1, 2, 3, 4, and 5]. Some proxy signature schemes are constructed from pairings [6, 7]. A few proxy signature schemes are constructed based on factoring problems. None of the above mentioned proxy signature schemes have the instantaneous revocation capabilities.

In the proposed schema, four entities are involved, Original Signer (S), Delegated entity (D), Trusted Delegation Authority (T), and a Verifier (V). The protocol produces a proxy signature on a message m. Basically; the original signer can delegate his signing right to a designed proxy signer to sign a message on behalf of the original signer. Then, a verifier, which knows the public keys of the original signer and the proxy signer, can verify the validity of the proxy signature by a proxy signer. For instance, a manager may need to delegate his secretary to sign messages on behalf of him. However, it is dangerous for him to give his private key to his secretary. A proxy signature scheme provides a method, by which the original signer authorizes a designated person, called delegated entity, to sign messages on behalf of him. In some countries, it may be required for the delegated entity to sign on behalf of the original signer, a trusted third party must approve this delegation by giving warrant that specifies kind of message are delegated and delegation period. We suggest a trusted party called delegation authority who witnesses that the original signer has been delegated the delegated entity. The proposed scheme is divided into the following stages: 1) Setup Stage is executed once, 2) Signing Stage is done whenever D signs a message m on behave of S, 3) Verifying Stage is done when V verifiers the signature of S on m, 4) Revocation Stage is done once when S wants to end his delegation for D.

Manuscript received March 5, 2011
Manuscript revised March 20, 2011

2. Preliminaries

Let E={ S, D, T, V} be the set of entities involved where S designates the original Signer, D designates the delegated entities (Proxy Signer), T for the Trusted Delegation Authority that witnesses that S has delegated D, and finally V represents the Verifier. Let X Є E, X is represented as follows. X= (pubx, $priv_x$, n_x), pub_x is the public key of Entity X, $priv_x$ is the private key of X, and n_x is the RSA constant. Recall that pub_x and privx are multiplicative inverses to each other mod Φ (n_x), where Φ (.) is the Euler totient function. We assume $priv_x$ Є {MAX (p_x,q_x), Φ (n_x)-1} is a large prime for two large primes p_x,q_x, and n_x = p_x * q_x. We also assume for message M the following holds:

$(M^{pubx}) \bmod n_x)^{privx}) \bmod n_x =$
$(M^{privx}) \bmod n_x)^{pubx}) \bmod n_x = M.$

We also assume for X Є E, X does not knows $priv_y$ of Y Є E, Y ≠ X but knows the rest of Y parameters and DC_x denotes the digital certificate of X Є E. Thus we can represent the entities involved as follows:

1. S= (pub_s, $priv_s$, n_s).
2. D= (pub_d, $priv_d$, n_d).
3. T= (pub_T, $priv_T$, n_T).
4. V= (pub_v, $priv_v$, n_v).

For two entities X, Y Є E, Y ≠ X, a message m from X to Y is sent over a secure channel that follows the secure socket protocol as follows:

X signs on m, generating encrypt $_{privx}$(H(m)) where H is a hashing one way function, X generates a session key SK, X generates the encrypted message encrypt $_{SK}$ (m ||encrypt $_{privx}$(H(m))|| DC_x), X generates the digital envelop encrypt $_{puby}$(SK) to achieve privacy, and finally X sends both the encrypted message and the digital envelop to Y. Y opens the envelop as follows: SK= decrypt $_{privy}$ (encrypt $_{puby}$(SK)), Y gets m ||encrypt $_{privx}$(H(m))|| DC_x =decrypt$_{SK}$ encrypt $_{SK}$(m ||encrypt $_{privx}$(H(m))|| DC_x), Y verifies H(m) = decrypt$_{pubx}$(encrypt $_{privx}$(H(m)) achieving authenticity, non repudiation, and message integrity.

3. The Proposed Protocol

3.1 Setup Stage

The setup stages involve 5 messages as shown in Figure 1.

3.1.1 Delegation Request (S to T)
S generates a Delegation Request message DReq and sends that request to T. DReq contains a warrant wm, which records the delegation policy including valid period, authority limitations, message type to be signed by the proxy signer.

3.1.2 Approval Request (T to D)
T verifies the signature of the original signer S on DReq. T approves this request by generating Approval Request message AReq and sends this message to D. AReq message contains the request of S for D to be his proxy signer, the context of delegation, the signature of S on this request, and the witnessing of T on that request by signing AReq message.

3.1.3Approval Response (D to T)
D approves the delegation from S, the context of delegation, and the witnessing of T on the delegation. D generates ARes message that include his acceptance for the delegation and the delegation terms.

3.1.4 Delegation Response (T to S)
T generates Delegation Response message DRes that contains the approval and the acceptance of the delegation and the witnessing of T on that delegation.

3.1.5 Blinded Signature Key message (S to D)
The only assumption we make is to consider privs is a large prime.
S performs the following.

a. S generates large prime N1Є {MAX (p_s,q_s),Φ(n_s)-1 }. S computes N2 the inverse of N2 Mod Φ (n_s) for two large primes p_s,q_s, and n_s= p_s* q_s .
b. S Creates the blinded signature N3=N1 * priv$_s$, Thus N3 is as large prime which is hard to factorize.
c. The blinded signature key N3 equals the multiplication of two large primes which is hard to factorize. S wants to send N3 to D to use in signing on behave of him without the knowledge of the original signing key priv$_s$.
d. S Signs on (N3, N2) by his private key, creates M1= (N3, N2) ^{privs} Mod (n_s).
e. S creates an envelope BSK= (M1) ^{pubD} Mod (n_D), where BSK is a Blinded Signature Key message.
f. S sends BSK to D, D retrieves N3, N2 as follows:
   I. M1= (BSK) ^{privD} Mod (n_D).
   II. (N3, N2)== (M1) ^{pubs} Mod (n_s).

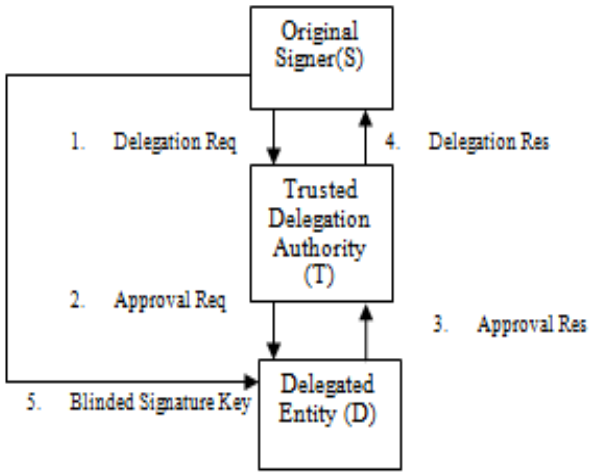

Figure 1: Messages exchanged during the Set up Stage

3.2 The Signing Stage

The signing stage involves three messages as shown in Figure 2.

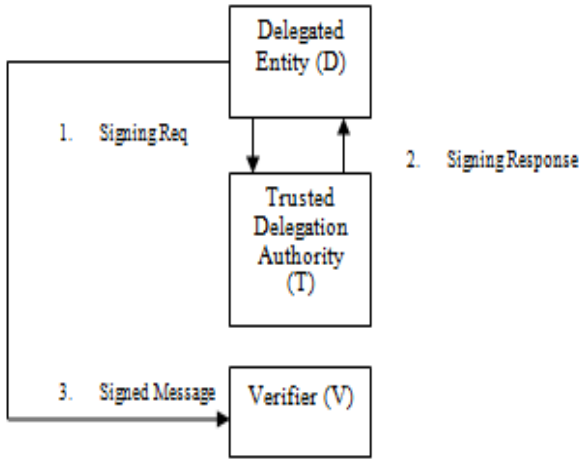

Figure 2: The Signing Stage

3.2.1 Signing Request (D to T)

Assume D wants to sign a message M on behalf of S. D generates a signing request message SReq and send it to T on a secure channel as explained in preliminaries section above.

3.2.2 Signing Response (T to D)

When T receives SReq, T retrieves M and checks that M is within the context and the authority of D as delegated from the original signer S, If D is dedicated with the policy of S, T will send a signing response to D approving for the signature by signing on M generating Signing Response message SRes, SRes contains the signature on M by T; S1= (H (M)) privTMod(nT).

A couple of points worth discussion.

- We assume request / response is based on Remote Procedure Call (RPC) mechanism which requires a permanent on line connection between the requester and the Responder. Since this is required in every transaction, the overall performance may be degraded down.
- We propose using mobile agent paradigm [8] and run time verification [9] only for the signing stage. The Signing request message is carried out by a mobile agent generated by host D, Dispatched from D and arrives at T, and the communication line is disconnected between D and T as shown in Figure 3. T performs a run time verification to assure that D is dedicated with the delegation policy of the original signer S. Then the mobile agent carries the signing response from T and returns back to D.

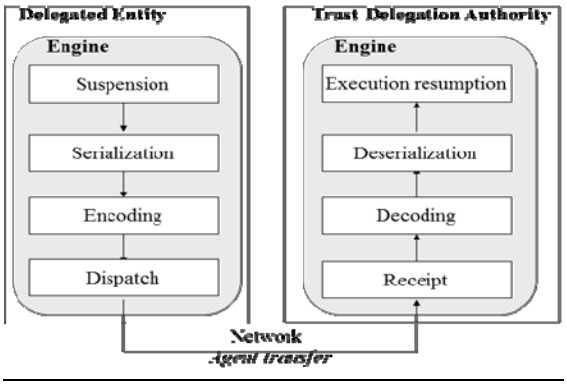

Figure 3: The Mobile Agent

- T to perform real time verification on M, A standard reference language for writing M is required. In that regard we assume using universal standard domain ontology. and ontology engineering [10] for that purpose.

3.2.3 Signed Message

D will verify the signature of T and add blind signature:
S2= (S1) privDMod (nD).
S3= (S2) N3 Mod (ns).
D will send S3 and N2 to V through secure channel as explained in preliminaries section above

3.3 Verification Stage

V verifies the signature of D, T, and S on M as follows:

a. $S2=(S3)^{N2}$ Mod(n_s)= $((S2)^{N1*privs}$ Mod $(n_s)^{N2}$ Mod (n_s) = $(S2)^{privs}$Mod(n_s).
b. $S1= (S2)^{pubD}$ Mod (n_D).
c. H (M) = $(S1)^{pubT}$ Mod (n_T).

Thus V verified the following:

- The message is coming from delegated entity (proxy signer), the delegated entity is delegated from signer, the delegation authority approves on this delegation, and the message is signed by original signer.
  T will notify S If D violates his delegation policy. In this case, the signing response message as a response for the signing request message from D will be a rejection and T will refuse to sign on M. T will send a notification message to S, S will send a Revocation Request message RReq and Receives A Revocation Response RRes Message as will be shown in the next section.

3.4 Revocation Stage

Figure 4 shows the messages incorporated in the revocation stage.

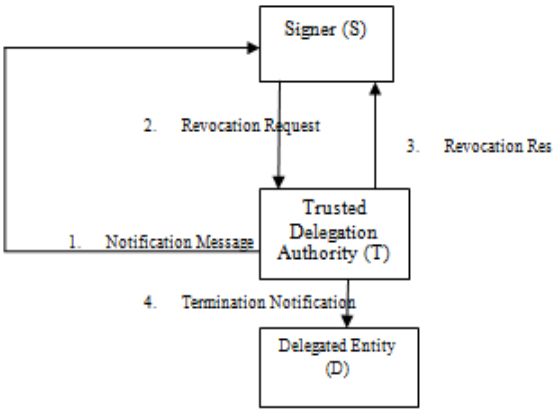

Figure 4: The Revocation Stage

3.4.1 Notification Message (T to S)
When T detects that D violates the delegation policy as dedicated by S, T sends a notification message to S to inform him by this violation.

3.3.2 Revocation Request (S to T)
Based upon the notification received from T, S sends a Revocation Request message RReq.

3.3.3 Revocation Response (T to S)
T replies to RReq by a Revocation Response message RRes to S. By RRes the delegation of S to D does no longer exist.

3.3.4 Termination Notification
T sends a termination notification message to D to inform him that the delegation of S to him is terminated.

Conclusion and Future Work

The basic advantage of the proposed protocol is as follows. The protocol is a fully controlled delegation protocol with instantaneous revocation capabilities. Using domain ontology for expressing delegation policy and run time verification for this policy deserves more investigation. The proposed scheme allows easy, simple, and instantaneous revocation. The proposed scheme suggests using ontology to allow real time verification for the message signed by the delegated entity (Proxy Signer) and assure the compliance of the proxy signer with the delegation policy dedicated by the original signer.

REFERENCES

[1] Kim S., Park S., & Won D. (1997) Proxy signatures. Revisited. In: ICICS'97. LNCS 1334.Springer-Verlag, 223–32.
[2] Sun H-M, Lee N-Y, & Hwang T. (1999). Threshold proxy signatures. IEE Proc – Comput Digit Tech, 146(5), 259–63.
[3] Sun H-M. (2000). Design of time-stamped proxy signatures with traceable receivers. IEE ProcComput. Digit. Tech., 147(6), 462–6.
[4] Yi L, Bai G, & Xiao G. (2000). A new type of proxy signature scheme. Electron Lett, 36(6) 527–8.
[5] Zhang N, Shi Q, & Merabti M. (2000). Anonymous public-key certificates for anonymous and fair document exchange. IEE Proc – Comput Digit Tech, 147(6), 345–50.
[6] Zhang F, Safavi-Naini R, & Lin C-Y. (2003). New proxy signature, proxy blind signature and proxy ring signature schemes from bilinear pairing. In: Information security and privacy (ACISP'03), LNCS 2727. Springer-Verlag, 312–23.
[7] Okamoto T, Inomata A, & Okamoto E. (2005). A proposal of short proxy signature using pairing. In: Information technology: coding and computing. ITCC, 1(4–6, 631–5.
[8] Outtagarads A Mobile Agent _ Based Applications: A Survey IJCSNS International Journal of Computer Science and Network Security, Vol 9 No 11, November 2009.
[9] Chen F, and RosuG An Efficient and Generic Runtime Verification Frame Work OOPSLA' 07, ACM Press, pp569 - 588, 2007
[10] Gandan Ontology Engineering: A Survey and Return of Experience Institute De Research en Informatic ET Auomatique Report no 4396, March 2002.

Khaled Shehata received his BSc from Military Technical College, Cairo, Egypt in 1981. After working as a research assistant he got his MSc. from Cairo University, Egypt in 1991. He received his PhD. from Naval Postgraduate School, Monterey, California, USA in 1996. He worked as a researcher in Egypt, then a Director for the VLSI design center, AOI, Egypt, finally he is a professor in the Arab Academy for Science and Technology, College of Engineering since 2000 till now. His research interests include VLSI design, electronic system design and have more than 70 scientific research papers in these areas.

Dr.Maged Elgendy received the B.Sc , ME, and PHD degrees from Ain-Shams University (Cairo, Egypt), Faculty of Science, Department of Mathematics. His research includes Cryptographic Engineering, , Zero-Knowledge Interactive Proofs Systems, and Public-Key Infrastructure (PKI). He is the executive Director of the Government Electronic Certification Authority in Egypt and the Advisor of the Egyptian Financial Minister.

Mohamed Mohamed Kouta received his BSc degree in Computer Engineering from Arab Academy for Science and Technology. He worked as developer in Ministry of State for Administration Development. Currently he is a developer & a database administrator in Government Certificate Authority (Gov CA) Project in the Egyptian Ministry of Finance.

Gamal Selim received his BSc from Military Technical College (M.T.C.), Cairo, Egypt in 1978. He was working as a teaching assistant from 1978 to 1980 at M.T.C. He got his MSc. and PhD. from Ohio State University, USA in 1982 and 1985 respectively. He was a staff member at M.T.C since 1985 till 1999. He joined the Arab Academy for Science and Technology, College of Engineering since 2000 till now . His research interests include digital system design, microprocessor and microcontroller based system.