The Efficiency and Security of Pseudonymous Digital Signature for Modern e-Services

Faiz Ahmad, Rajesh Jalnekar

Department of Computer Engineering, Bharati Vidyapeeth University , Pune-411043(India)
Department of Electronic and Telecommunications, Vishwakarma Institute of Technology, Pune-411037(India)

Summary
This paper presents a cryptanalysis of Pseudonymous Digital Signature(PDS) described in [8]. PDS proposed with respect to Pseudonymous Identification Scheme

Intended to generate pseudonyms by trusted third party (usually the pseudonymity service provider) for registered users and organizations that provides on –line services. The results shows that the PDS very fast and highly secure with a comparison to RSA signature. Furthermore PDS offers a powerful mechanism to overcome the drawbacks that a pseudonymous credentials suffering to enable secure and flexible modern e-services.

Key words :

Pseudonym, Pseudonymous Signature, Security.

1. Introduction

Pseudonyms are identifiers of subjects. The subject that may be identified by the pseudonym is the holder of the pseudonym. An essential factor for effectiveness of pseudonyms is the unlinkability between the pseudonym and its holder and if pseudonyms can be linked between each other[6].

In practical terms, anonymity occurs when a user's identity cannot be ascertained. An example of an anonymous transaction is one in which neither participant recognises or knows anything about the other. A significant disadvantage of anonymity is that accountability becomes problematic and therefore anonymity services are exploitable by those engaged in criminal activities. The highest degree of anonymity can be reached with little knowledge of the linking between the holder of a pseudonym and its pseudonym.

Pseudonymity provides a compromise between anonymity and accountability. A user employing a pseudonym engages in communications and transactions without revealing their identity. The link ability between the holder and his pseudonym may be known to third parties or only to the holder of the pseudonym.

In most existing credentials systems a pseudonymous certificate binds a user's pseudonym to their public key, the private key to which the user possesses. Such certificates are issued by a trusted provider. Identities, pseudonyms and public keys should be unique. Based on how pseudonyms are generated, there are two ways to generate globally unique pseudonyms for a person (here called holder)[7]:

Centralized Generation:
This approach employs a centralized third party, which generates the pseudonym on the user's behalf. This party can easily avoid duplicates and hence the generated pseudonyms are unique. Additionally, the holder of the certificate has to trust in the issuer, since the issuer knows the linking of the holders identity to his pseudonym.

Local (Holder-based) Generation:
The other way is, that the user generates his pseudonym locally. Now, only the user knows the linking between his identity and his pseudonym.

In the approach presented in this paper, the centralized third party generates for every registered player a globally public unique pseudonym and related private pseudonym while the player updates his related private pseudonym whenever needed locally in his personal security environment. The pseudonymous certificate uniquely identifies by public unique pseudonym and it contains no binding between a public pseudonym and the name of it's holder.

An organization that provides e-services can pseudonymously prove possession of shared secret keys with an user willing to grant access to such service simply by implementing pseudonymous digital signature (PDS). The PDS scheme provides a strong mechanism to prove pseudonymous and accountable membership of trusted domain for both the users and organizations.

The rest of this paper is organized as follows: The next section describes the generation of pseudonyms. Subsequently, Section 3 explains the pseudonymous digital signature. Sections 4 describes implementation of PDS algorithm. Finally, the paper concludes in Section 5.

2. Generation of Pseudonyms

The method presented in this paper to generate pseudonyms is based on the Pseudonymous Identification Scheme(PIS) presented in[8]. PIS rely on the following assumption:

Assumptions. *for any trusted centre with an integrated RSA modulus* $n \in Z_n^* : n = p \cdot q \cdot f$ *(p, q and f are three prime numbers with approximately of k-bits length each) and the related integrated Euler's totient function* $\phi(n)$ *=(p-1)(q-1)(f-1), it is always possible to generate a secret random integer* $r \xleftarrow{R} Z_{\phi(n)}$ *, and the related public integer* $d \in Z_{\phi(n)}$ *such that r :* $2 < r < \phi(n)$ *and* $d = \phi(n)$*-r, where* $\exists g \in Z_n^*$ *and the following conditions must be hold:*

$$\phi(n)^{\phi(n)} \equiv 1 \bmod n \text{ and } g^{\phi(n)} \equiv 1 \bmod n \qquad (1)$$

According to this assumptions every player (e.g. users and organisations) registered with TTP will obtains pseudonymous certificate uniquely identified by *public integer d* as a cryptographic unique public pseudonym denoted by pseudonym ID_p, while the *secret integer r* considered as a cryptographic private pseudonym denoted by ID_v. A player's unitary pseudonymous identity denoted by ID_u*:*

$$ID_u = ID_p + ID_v = k \cdot \phi(n)$$

Where *k*-random integer number.

It is assumed that no two player registered with a trusted third party have the same public or private pseudonyms under the unique integrated RSA modulus.

More generally, a pseudonym ID_v is generated by use of a function *f* parameterized with two parameters: the player's unitary pseudonymous identity ID_u and a unique public pseudonym ID_p. Hence the pseudonym ID_v results in:

$$ID_v = f(ID_u, ID_p) = k \cdot \phi(n) - ID_p$$

More precisely, the TTP generates unique pseudonym ID_p for the player while the player responsible for updating private pseudonym ID_v in his personal security environment such that:

$$ID_v = f(C \cdot ID_u, ID_p) = C \cdot ID_u - ID_p \qquad (2)$$

Where *C*-random update integer number generated by a player.

There is no need for any global data or information interchange between issuing party and updating parties. The only requirement is that in addition to pseudonymous certificate the player grants a secret 2-tuple(ID_v ,*g*) upon completion his registration phase with TTP.

3. Pseudonymous Digital Signature(PDS)

In this section we provide a brief description of PDS scheme with some new enhancement and then perform an efficient cryptanalysis on algorithm comparing it to RSA signature.

3.1 Parameter Generation

let G be a finite cyclic group, and let *g* be the generator of prime order *n* in *G* , a trusted third party (TTP) generates an *integrated RSA modulus* $n \in Z_n^*$ (chooses L_n = 2048 bits or above) and the related *Euler's totient function* $\phi(n)$. After computing values of n and $\phi(n)$ a trusted authority chooses a generator $g \xleftarrow{R} G^*$, then checks if $\phi(n)$ and generator *g* are satisfying the conditions (1). If they do then it considers *(n,* $\phi(n)$ *, g)* as a system wide parameters.

The pseudonyms for the players (e.g. users and organisations) are generated according to the manner explained in previous section.

3.2 Signing Algorithm

Consider the protocol is a session between a user $U(UIP_P, UID_V)$ wants to access some service with an organization $O(OIP_P, OID_V)$.

Sign(UID_V , OID_P , *g* ,*M*)**.** This algorithm takes as input a signer's secret pseudonym(user) UID_V , a destination's public pseudonym(organization) OID_P, generator *g*, and a message $M \in \{0,1\}^*$ and proceeds as follow:

(i) The user computes the sum of his secret pseudonym UID_V and organization's public pseudonym OID_P and generates what we called *encryption shared secret key* E_{SSK} :

$$E_{SSK} \equiv (g^{UID_V + OID_P}) \bmod n \qquad (3)$$

(ii) A user then encrypts the message *M* using E_{SSK} and sends it to organization that provides such service:

$$S_U(M) \equiv (M \times E_{EEK}) \bmod n \qquad (4)$$

3.3 Verification Algorithm

Verify($S_U(M)$, OID_V, UID_P , g). The verification algorithm takes as input signer's public pseudonym UID_P , a verifier's secret pseudonym OID_V ,generator *g*, and a purported signature $S_U(M)$, and proceeds as follow :

(iii) The organization computes the sum of its secret pseudonym OID_V and user's public pseudonym UID_P , then generates *a decryption shared secret key* D_{SSK} :

$$D_{SSK} \equiv (g^{OID_V + UID_P}) \bmod n \qquad (5)$$

(iv) The organization decrypts the massage using D_{SSK} :

$$M \equiv (S_U(M) \times D_{SSK}) \bmod n \qquad (6)$$

If the message *M* equal to decrypted one the signature is accepted and the organization ensured that the signer belongs to trusted pseudonymous user from the same trusted domain, then the user is granted access to the intended service.

4. Implementation of PDS Algorithm

Key generation time, signature time, and verification time are all indicators of a signature scheme's performance. However, no one aspect alone is enough to judge whether one signature scheme is better than another for all situations.
Given a set of parameters the constitute the algorithm , We define some metrics using these parameters which compute a single amortized cost for the performance, allowing us to make direct comparisons between PDS and RSA schemes for any given situation.

4.1 Time Analysis

4.1.1 Timings for PDS With N= 8192-bit

All the times recorded in (table.1, table.2) have been measured on a AMD Turion(tm) 64x2 Mobile technology TL- 50 1.60 GHz processor, using the time measurement functions offered by the Java library on a Windows Vista platform. From the table.1, the key generation time for PDS with modulus N= 8192 bits : 18392 Milliseconds (averaged over 3 samples). Figure1, shows the change of encryption/decryption time when the message length varying between(512-12288 char) with the same encryption/decryption shared secret keys generating between two parties (e.g. users and organisations).

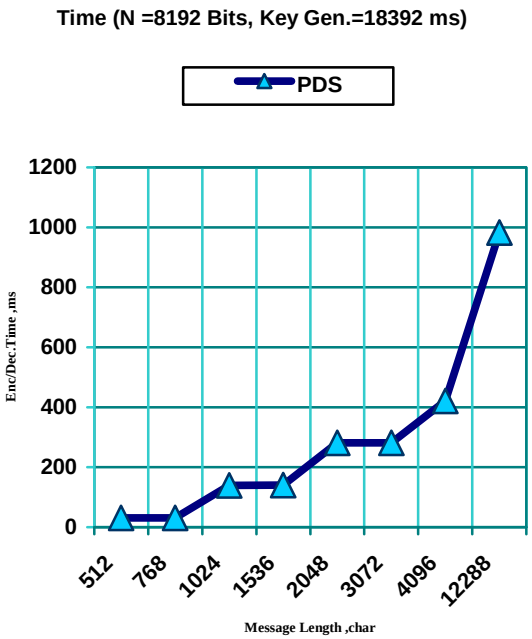

Figure1, PDS Encryption/Decryption Time with N=8192 bits

4.1.2 Timings for RSA With N= 8192-bit

From the table.1, the Key generation time for RSA with modulus N=8192 bits : 251565 Milliseconds (averaged over 3 samples). Figure2, explains the change of encryption/decryption time when the message length varying between(512-12288 char) with the same encryption/decryption shared secret keys generating between two parties

The results show the PDS Encryption/Decryption Time is trivial compared to RSA Encryption/Decryption Time with various message sizes.

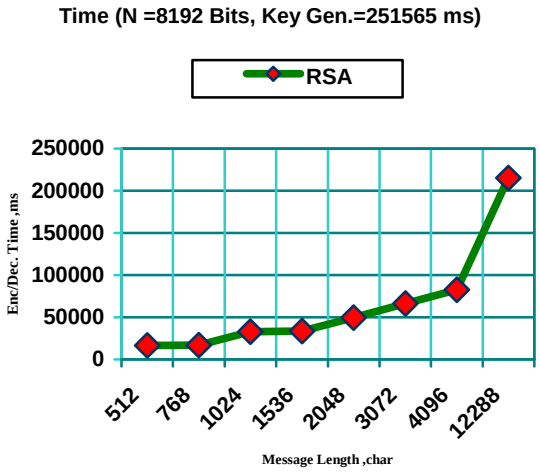

Figure2, RSA Encryption/Decryption Time with N=8192 bits

PDS		Block size = 8				N=8192		
Key Gen. time, MS	18392							
Message length, char	512	768	1024	1536	2048	3072	4096	12288
Encryption/decryption time, MS	31	31	140	141	281	281	421	983
RSA		Block size = 8				N=8192		
Key Gen. time, MS	251565							
Message length, char	512	768	1024	1536	2048	3072	4096	12288
Encryption/decryption time, MS	16520	16863	33009	33680	49671	66284	82742	215296

Table.1, shows the tests results on PDS and RSA with constant N= 8192 bits and various messages lengths.

PDS		Message length = 12288 char			Block size = 8		
Modulus N, bits	1024	1536	2048	3072	4096	6144	8192
Key generation time, MS	296	390	671	1404	3244	8175	18392
Encryption/decryption time, MS	437	452	592	718	742	843	983
RSA		Message length = 12288 char			Block size = 8		
Key length	1024	1536	2048	3072	4096	6144	8192
Key generation time, MS	624	1217	3806	15475	43274	279131	474552
Encryption/decryption time, MS	4633	8283	14461	31169	53912	120245	214828

Table.2, shows the tests results on PDS and RSA with various keys lengths and the same input file (12288 char).

4.1.3 Key Generation Time for PDS and RSA with Different Modulus Lengths

The table.2 shows the key generation times for PDS and RSA schemes which were recorded with various bit-strengths. Here we have to mention that for PDS algorithm the key generation time not includes the time of generation of modulus N, because it generated only once by a trusted third party. Figure 3, describes the curve of key generation time when the lengths of modulus changed. As it seem from the figure 3, the PDS key generation time is very short and approximately same for all tested values of modulus N, while RSA key generation time with same tested lengths is very long and it is increased accordingly to the size of N.

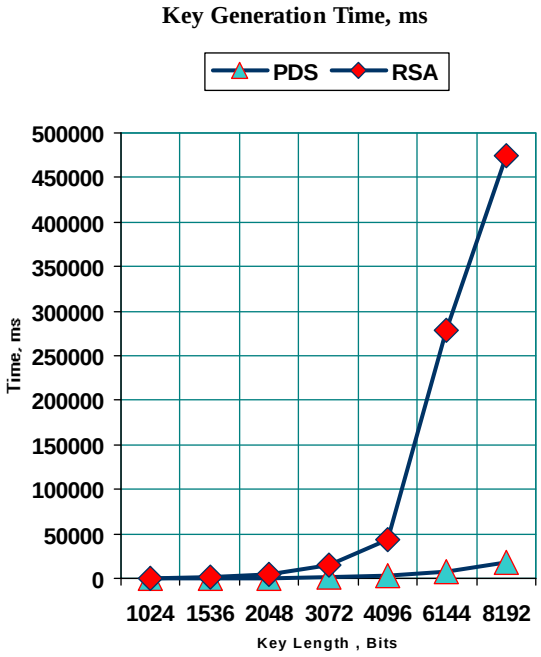

Figure 3, the key generation time with various bit-strengths of modulus N.

4.1.4 Encryption/Decryption Time for PDS and RSA With Input File (12288 char)

Figure 4, describes the curve of encryption/decryption time with the same input file (12288 char) and various key lengths as it recorded in table.2.

For RSA algorithm the 1024-bit RSA is definitely the fastest among the ones shown, in term of key generation and encryption/decryption times, it is not the most secure, providing marginal security from a concerted attack. The slowest (4096-bit RSA and above) should be used in very critical situations since it offers the maximum resistance to attacks but with high cost. In our opinion the 2048-bit modulus is a good balance between speed and security for RSA with acceptable cost.

While PDS algorithm with all tested modulus offers approximately the same degree of efficiency in term of key generation and encryption/decryption times. In our opinion the 4096-bit PDS modulus is a good balance between speed and security with very low cost. The 8192-bit PDS offers the maximum resistance to attacks with suitable cost and it also could be an attractive applicable solution for some applications.

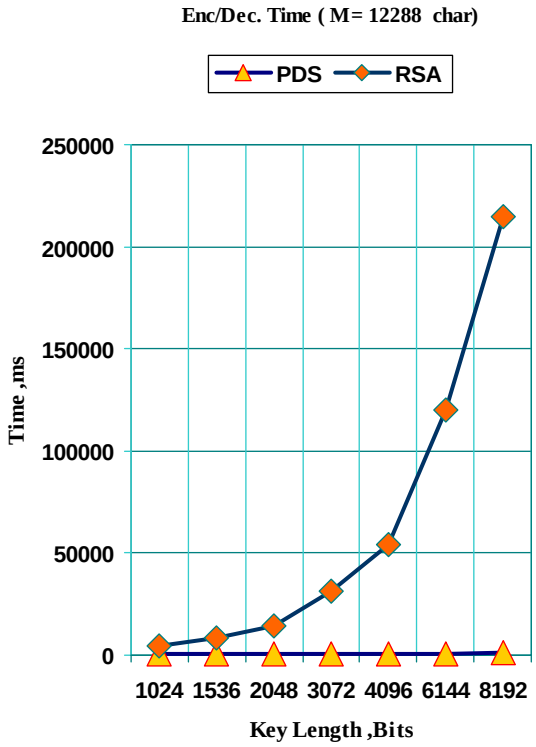

Figure 4, encryption/decryption time with the same input file (12288 char)

4.2 Security Analysis

The 4096-bit PDS modulus is significantly large to be factorized. The factorization of PDS modulus is known only and only by trusted third party even it is shared among all registered parties with TTP.

In addition to hard factorization the PDS algorithm is secure against timing attacks. Timing attacks are possible whenever an operation is performed in an automated and interactive fashion, such as protocol negotiation or operations performed by a smart card. By choosing specific inputs and measuring the time between request and reply, it is possible for an attacker to infer information about the private key that compromises security[12]. Due the short key generation time for every communication session and the fast encryption/decryption time, an attacker can't infer any information about the shared secret keys. The only threat comes from the forgery of user's private pseudonym UID_V .

4.2.1 Forgery of a User's Private Pseudonym (Impersonation)

The problem of pseudonyms presented in this paper is that, a private pseudonym which has been disclosed to a verifier may be used by the verifier to impersonate its original holder. This is possible in case TTP permit identical unitary pseudonymous identity $ID_u = k \cdot \phi(n)$ for all registered members. So after disclosure the verifier knows UID_P and UID_V for an user . Hence he can act like the original holder; he may use and disclose the 'stolen' pseudonym to proof the ownership, which enables him to impersonate the original holder.

A straight-forward solution for this problem is to make unitary pseudonymous identity different for each registered player with TTP. In addition, the player has ability to prevent forgery of his private pseudonym simply by updating it locally in his personal security environment according to formula(2).

5. Conclusion and Future Work

In this paper we presented a cryptanalysis of Pseudonymous Digital Signature(PDS) which based on a scheme for generating digital pseudonyms, that apply a basic centralized issuer and consequently distributed updaters. The test results shows that the PDS very fast and highly secure at the low cost as compared to RSA signature.

PDS offer a flexible and scalable solution for access control in modern e-services where the pseudonymity, speed and security much desirable. On the one hand, incorporating PDS, digital certificates and trustee services, greatly enhance the privacy of individuals involved in such pseudonymous environment.

References

[1] R.L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Communications of the ACM, vol. 21, no. 2, pp 120-126, February 1978.

[2] T. ElGamal, "A public-key cryptosystem and signature scheme based on discrete logarithms," In Crypto '84, volume 196 of LNCS, pp. 10–18, Berlin, Springer-Verlag, 1985

[3] Dan Boneh and Matthew Franklin. ," Efficient Generation of Shared RSA Keys," Journal of the ACM, Vol. 48, No. 4, July 2001.

[4] Gilboa N,"Two party RSA key generation," In *Advances in Cryptology—Crypto '99*. Lecture Notes in Computer Science, vol. 1666. Springer-Verlag, New York, pp. 116–129,1999

[5] Aleksandra Nenadiæ, Ning Zhang, Barry Cheetham, Carole Goble," RSA-based Certified Delivery of E-Goods Using Verifiable and Recoverable Signature Encryption," Journal of Universal Computer Science, vol. 11, no. 1 (2005), 175-192

[6] Oliver Jorns, Gerald Quirchmayr, Oliver Jung," A Privacy Enhancing Mechanism based on Pseudonyms for Identity Protection in Location-Based Services"
Australian Computer Society, Vol. 68, 2007, pages 133-142.

[7] Peter Schartner and Martin Schaffer," Unique User-generated Digital Pseudonyms" Originally published in Springer LNCS 3685.

[8] Faiz Ahmad, Rajesh Jalnekar," Modern Credential Access Control Approach Based On Pseudonymous Signature", IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.10, October 2007, pages 129-134.

[9] E. R. Verheul. Self-blindable credential certificates from the Weil pairing. In C. Boyd, editor, Proceedings of Asiacrypt 2001, volume 2248 of LNCS, pages 533–51. Springer-Verlag, Dec. 2001.

[10] B. Waters. Efficient identity-based encryption without random oracles. In R. Cramer, editor, Proceedings of Eurocrypt 2005, volume 3494 of LNCS, pages 114–27. Springer- Verlag, May 2005.

[11]S. Micali and R. Rivest. Transitive signature schemes. In B. Preneel, editor, Proceedings of CT-RSA 2002, volume 2271 of LNCS, pages 236–43. Springer-Verlag, Feb.2002.

[12] Eric Cronin, Sugih Jamin, Tal Malkin, Patrick McDaniel," On the Performance, Feasibility, and Use of Forward Secure Signatures," ACM October 27–31, 2003, pages 131-144.

[13]A. Pashalidis and C.J. Mitchell, A Security Model for Anonymous Credential Systems, Proceedings of the 3rd Working Conference on Privacy and Anonymity in Networked and Distributed Systems (I-NetSec'04), Kluwer Academic Publishers, pages 183-199, Toulouse, France, August 2004.

[14] J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT 2001*, *LNCS 2045*, pages 93–118, 2001.

[15] B. Friedman, P.H. Khan, and D.C. Howe. Trust Online. In Communications of the ACM, volume 43, pages 34–40, 2000.

[16] Richard Au, Harikrishna Vasanta, Kim Kwang Raymond Choo, Mark Looi. A User Centric Anonymous Authorisation Framework in Ecommerce
Environment. ICEC'04, Sixth International Conference on Electronic Commerce, ACM, pages 138-147.

[17] S. Chow, C. Boyd, and J. Gonzalez. Security-mediated certificateless cryptography. In *PKC 2006*, volume 3958 of *Lecture Notes in Computer Science*, pages 508-524. Springer-Verlag, 2006.

[18] Kohlas and U. Maurer, Reasoning about public-key certification - on bindings between entities and public keys, IEEE JSAC, vol. 18, no. 4, Apr, 2000.

[19] B. Pftzmann. Privacy in enterprise identity federation - policies for Liberty 2 single sign on. *Information Security Technical Report*, 9(1):45-58, January-March 2004.

F**aiz ahmad** received the B.E. and M.E. degrees in Automation in 1997. He received the M.E. degree in computer science and engineering from Pune Univ. in 2004. Since 2005 he is a Ph.D candidate at Bharati Vidyapeeth University. His research interest cryptography and information's security.

Rajesh Jalnekar received the B.E. , M.E. and Ph.D degrees in Electronic and telecommunication engineering from Univ. of Pune. His research interest network security and signal processing. he is working as dean academic development at VIT Pune.