INFORMATICA, 2006, Vol. 17, No. 2, 177–186
© 2006 *Institute of Mathematics and Informatics, Vilnius*

Identity Based Multisignatures [*]

Raju GANGISHETTI, M. Choudary GORANTLA, Manik Lal DAS,
Ashutosh SAXENA
Institute for Development and Research in Banking Technology
Castle Hills, Road No.1, Masab Tank, Hyderabad, India
e-mail: {graju, gmchoudary}@mtech.idrbt.ac.in, {mldas,asaxena}@idrbt.ac.in

Received: May 2005

Abstract. This paper presents identity based serial and parallel multisignature schemes using bilinear pairings. Our serial multisignature scheme requires a forced verification at every level to avoid the overlooking of the predecessors' signatures. However, in parallel multisignature scheme the verification of individual signatures is performed by a designated clerk. We show that our schemes are secure against existential forgery under adaptive chosen message attack in the random oracle model.

Key words: serial multisignatures, parallel multisignatures, bilinear pairings, identity based cryptosystems.

1. Introduction

In day-to-day life, many legal documents require signatures from more than one party, e.g., contracts, decision making processes, petitions etc. To meet these requirements in the digital environment, cryptography provides a mechanism known as multisignature. A multisignature scheme provides:

- multiple signers to generate a signature for a single message,
- a convincing mechanism to the verifier that each stated signer had signed the message.

A multisignature scheme is practical when the size of the multisignature by n signers is less than the total size of n signatures in the single signature scheme, on which the multisignature scheme is based. Accordingly, the verification cost gets reduced.

Based on the nature of applications, the multisignatures have been categorized into two types: *serial* and *parallel*. In serial multisignature, a signer signs the message and sends it to the next signer for further processing; the next signer after verifying his predecessor's signature, signs the received component. The serial multisignature generation is considered to be complete when the last signer signs. Many financial transactions require serial multisignatures and verification at each level. For e.g., in the maker-checker-approval concept, where maker prepares the transaction and checker ensures the correct-

[*]This work is supported in part by the Ministry of Communications and Information Technology, Government of India, Under the Grant No. 12(35)/05-IRSD.

ness of the transaction for approval. This process need to be followed in a sequence such that every signer is forced to verify his immediate predecessor's signature. In the case of parallel multisignature, the signature of each signer is carried out on the message itself but not on the signatures of the other signers. In order to complete the parallel multisignature generation, a designated clerk combines all the individual signatures after verifying them. Parallel multisignatures are useful in the organizations where a flat reporting structure exists.

Itakura and Nakamura (1983) introduced the concept of multisignature. Since then, several schemes (Okamoto, 1988; Boyd, 1989; Ohata and Okamoto, 1991; Harn, 1994; Horster *et al.*, 1995; Ohata and Okamoto, 1999; Lin *et al.*, 2001; Micali *et al.*, 2001; Boldyreva, 2003) for multisignatures have been proposed. The proposal of Harn (1994) was cryptanalyzed by Horster *et al.* (1995) and the scheme in (Ohata and Okamoto, 1991) avoids restriction on the signing order. It is noted that the security analysis of the scheme (Ohata and Okamoto, 1999) does not consider the key generation phase. A formal notion of security for multisignature was proposed by Micali *et al.* (2001). Later, Boldyreva (2003) proposed a generic notion of security for multisignature schemes. All the above schemes are proposed under certificate based public key cryptosystems. One may note that the traditional certificate based public key cryptosystems require large amount of storage and computing time to manage the certificate life cycle (Guttman, 2002).

Shamir (1985) introduced the concept of identity(ID) based cryptosystems where, a user's public key could be easily derived from his identity and the user's private key is generated by a trusted third party called Private Key Generator (PKG). ID-based cryptosystems are advantageous over the traditional public key cryptosystems (PKCs), as key distribution and revocation are simplified (Gorantla *et al.*, 2005). A verifier can verify a signature just by using the signer's identity. Lin *et al.* (2001) proposed ID-based structured multisignature scheme on which successful attack was carried out by Mitchell (2001).

In this paper, we propose ID-based serial and parallel multisignature schemes using bilinear pairings. To the best of our knowledge there is no existing secure serial ID-based multisignature scheme using pairings. We use Hess's ID-based signature scheme (Hess, 2003) as the base for our multisignature schemes. The schemes are secure against existential forgery under adaptive chosen message attack in the random oracle model assuming computational Diffie–Helman problem is hard.

The rest of the paper is organized as follows. In Section 2, we describe background concepts on bilinear pairings and some related mathematical problems. In Section 3 and 4, we present our proposed serial and parallel multisignature schemes respectively. Section 5 gives the security analysis of the proposed schemes. Finally, we conclude our work in Section 6.

2. Background Concepts

In this section, we briefly review the basic concepts on bilinear pairings and some related mathematical problems.

2.1. *Bilinear Pairings*

Let G_1 be an additive cyclic group of large prime order q, G_2 be a multiplicative cyclic group of the same order and P be a generator of G_1. A cryptographic bilinear map e is defined as e: $G_1 \times G_1 \rightarrow G_2$ with the following properties:

Bilinear: $e(aR, bS) = e(R, S)^{ab}$ $\forall R, S \in G_1$ and $a, b \in Z_q^*$. This can be restated as $\forall R, S, T \in G_1$, $e(R + S, T) = e(R, T)e(S, T)$ and $e(R, S + T) = e(R, S)e(R, T)$.

Non-degeneracy: If P is a generator of G_1, then $e(P, P)$ is a generator of G_2. In other words $e(P, P) \neq 1$.

Computable: There exists an efficient algorithm to compute $e(R, S)$ $\forall R, S \in G_1$.

In general implementation, G_1 is the group of points on an elliptic curve and G_2 denotes a multiplicative subgroup of a finite field. Typically, the mapping e is derived from either the Weil or the Tate pairing on an elliptic curve over a finite field. We refer to (Boneh and Franklin, 2001) for more comprehensive description on how these groups, pairings and other parameters are defined.

2.2. *Computational Problems*

Now, we give some computational problems, which will form the basis of security for our schemes.

Discrete Logarithm Problem (DLP): Given two elements $R, S \in G_1$, find an integer $a \in Z_q^*$, such that $S = aR$ whenever such an integer exists.

Computational Diffie–Hellman Problem (CDHP): For any a, $b \in Z_q^*$, given $< P, aP, bP >$, compute abP.

Decisional Diffie–Hellman Problem (DDHP): For any $a, b, c \in Z_q^*$, given $< P, aP, bP, cP >$, decide whether $c \equiv ab \bmod q$.

Bilinear Diffie–Hellman Problem (BDHP): For any $a, b, c \in Z_q^*$, given $< P, aP, bP, cP >$, compute $e(P, P)^{abc}$.

Gap Diffie–Hellman Problem (GDHP): A class of problems where CDHP is hard while DDHP is easy.

3. Proposed Serial Multisignature Scheme (SMS)

The proposed serial multisignature scheme consists of four phases: *Setup*, *Key Extraction*, *Multisignature Generation* and *Multisignature Verification*. The entities involved in our scheme are the Private Key Generator (PKG), set of Signers $\mathbf{S}$ and the Verifier $\mathbf{V}$.

3.1. *Description*

3.1.1. *Setup*

PKG publishes system parameters `params` which include $\{G_1, G_2, e, q, P, P_{pub}, H, h\}$, here G_1 is a cyclic additive group and G_2 is a cyclic multiplicative group with large prime order q. P is a generator of G_1, e: $G_1 \times G_1 \rightarrow G_2$ is a bilinear map between the groups

G_1 and G_2, H: $\{0,1\}^* \to G_1^*$ and h: $\{0,1\}^* \times G_2 \to Z_q^*$ where $G_1^* = G_1 \setminus \{0\}$ are the cryptographic hash functions. $P_{pub} = s_0 P$ is the public key of the PKG, where s_0 is master secret of the PKG and randomly picked from Z_q^*.

3.1.2. *Key Extraction*

Let I be the set of identities of n signers, $I = \{ID_1, ID_2, \ldots, ID_n\}$. For a signer with ID_i the public key is $Q_{ID_i} = H(ID_i)$. The PKG generates the private key as $S_{ID_i} = s_0 Q_{ID_i}$.

Note that the key extraction phase requires secure channel for the PKG to deliver private keys of the signers. This can be overcome efficiently by the secure issuing protocol proposed in (Gangishetti *et al.*, 2005).

3.1.3. *Multisignature Generation*

In this phase n signers with identities $\{ID_1, ID_2, \ldots, ID_n\}$ sequentially generate the multisignature and the final signer sends it to the verifier. To have a multisignature on message m, without loss of generality, we present it in following stages.

Signature Generation by First Signer

To sign a message m, the first signer picks a random integer $k_1 \in_R Z_q^*$ and computes

$$\begin{aligned} r_1' &= e(P,P)^{k_1},\\ r_1 &= r_1',\\ c_1 &= h(m, r_1),\\ u_1 &= c_1 S_{ID_1} + k_1 P. \end{aligned}$$

The signature by the first signer is the tuple $\langle u_1, c_1 \rangle$, which he sends to the second signer along with the message m.

Verification and Signature by Intermediate (ith) Signer

The ith signer verifies the signature $\langle u_{i-1}, c_1, c_2, \ldots, c_{i-1} \rangle$ received from $(i-1)$th signer by computing

$$r_{i-1} = e(u_{i-1}, P) e\Big(\sum_{j=1}^{i-1} c_j Q_{ID_j}, -P_{pub}\Big).$$

The signature is accepted if and only if $c_{i-1} = h(m, r_{i-1})$.

For generating his signature, the ith signer picks a random integer $k_i \in_R Z_q^*$ and computes

$$\begin{aligned} r_i' &= e(P,P)^{k_i},\\ r_i &= r_{i-1} r_i',\\ c_i &= h(m, r_i),\\ u_i &= u_{i-1} + c_i S_{ID_i} + k_i P. \end{aligned}$$

He then sends the partial multisignature $\langle u_i, c_1, c_2, \dots, c_i\rangle$ to the $(i+1)$th signer. One may note that ith signer cannot generate his signature without verifying the signature of $(i-1)$th signer, as it requires the extraction of r_{i-1} from the predecessor's signature $\langle u_{i-1}, c_1, c_2, \dots, c_{i-1}\rangle$. This ensures the forced verification.

Verification and Signature by the Final (nth) Signer

The nth signer verifies the signature $\langle u_{n-1}, c_1, c_2, \dots, c_{n-1}\rangle$ received from $(n-1)$th signer by computing

$$r_{n-1} = e(u_{n-1}, P)e\Big(\sum_{j=1}^{n-1} c_j Q_{ID_j}, -P_{pub}\Big).$$

The signature is accepted if and only if $c_{n-1} = h(m, r_{n-1})$.

For generating his signature, the nth signer picks a random integer $k_n \in_R Z_q^*$ and computes

$$\begin{aligned} r'_n &= e(P,P)^{k_n},\\ r_n &= r_{n-1}r'_n,\\ c_n &= h(m, r_n),\\ u_n &= u_{n-1} + c_n S_{ID_n} + k_n P. \end{aligned}$$

He then sends the final multisignature $\langle u_n, c_1, c_2, \dots, c_n\rangle$ along with the message m to the verifier.

3.1.4. *Multisignature Verification*

On receiving a signature $\langle u_n, c_1, c_2, \dots, c_n\rangle$ along with the message m, the receiver verifies the signature by computing

$$r_n = e(u_n, P)e\Big(\sum_{i=1}^{n} c_i Q_{ID_i}, -P_{pub}\Big).$$

The signature is accepted if and only if $c_n = h(m, r_n)$.

3.2. *Correctness*

The verification process of the multisignature $\langle u_n, c_1, c_2, \dots, c_n\rangle$ on message m is acceptable as given by the following equations:

$$\begin{aligned} &e(u_n, P)e\Big(\sum_{i=1}^{n} c_i Q_{ID_i}, -P_{pub}\Big)\\ &\quad = e\Big(\sum_{i=1}^{n}(c_i S_{ID_i} + k_i P), P\Big)e\Big(\sum_{i=1}^{n} c_i Q_{ID_i}, -P_{pub}\Big) \end{aligned}$$

$$
\begin{aligned}
&= e\Big(\sum_{i=1}^{n} c_i S_{ID_i}, P\Big) e\Big(\sum_{i=1}^{n} k_i P, P\Big) e\Big(\sum_{i=1}^{n} c_i Q_{ID_i}, -P_{pub}\Big)\\
&= e\Big(\sum_{i=1}^{n} c_i Q_{ID_i}, P_{pub}\Big) e\Big(\sum_{i=1}^{n} k_i P, P\Big) e\Big(\sum_{i=1}^{n} c_i Q_{ID_i}, -P_{pub}\Big)\\
&= e\Big(\sum_{i=1}^{n} k_i P, P\Big)\\
&= \prod_{i=1}^{n} e(P,P)^{k_i} = \prod_{i=1}^{n} r'_i = r_n.
\end{aligned}
$$

Thus the correctness of the scheme is justified.

4. Proposed Parallel Multisignature Scheme (PMS)

In parallel multisignature scheme the set of n signers with identities $\{ID_1,ID_2,\ldots,ID_n\}$ generate their individual signature on the given message m. In order to generate a parallel multisignature, a designated clerk, (typically one of the signers) collects, verifies and combines all the signatures. In this section we derive parallel multisignature scheme inheriting the concept of Chen *et al.* (2003).

The proposed parallel multisignature scheme consists of four phases: *Setup*, *Key Extraction*, *Multisignature Generation* and *Multisignature Verification*. The entities involved in our scheme are the Private Key Generator (PKG), set of Signers $\mathbf{S}$, a designated Clerk $\mathbf{C}$ and the Verifier $\mathbf{V}$.

4.1. *Description*

The two phases *Setup* and *Key Extraction* are same as given in the serial multisignature scheme.

4.1.1. *Multisignature Generation*

The parallel multisignature is generated as follows:

- Each signer with identity ID_i randomly selects an integer $k_i \in_R Z_q^*$, computes $r_i = e(P,P)^{k_i}$ and broadcasts r_i to the remaining $(n-1)$ signers.
- Each signer with identity ID_i computes $r = \prod_{i=1}^{n} r_i$ and $c = h(m,r)$, $c_i = h(m,r_i)$, $U_i = cS_{ID_i} + k_i P$. The partial signature on message m is $\langle U_i, c_i\rangle$.
- All the signers send their partial signatures to the clerk $\mathbf{C}$.
- Clerk verifies each individual signature by checking the equality

$$c_i = h\big(m, e(U_i,P)e(c_i Q_{ID}, -P_{pub})\big).$$

- Once all individual signatures are verified, $\mathbf{C}$ computes $U = \sum_{i=1}^{n} U_i$.

The parallel multisignature on message m is tuple $\langle U, c\rangle$.

4.1.2. *Multisignature Verification*

On receiving the multisignature $\langle U, c\rangle$ on message m the verifier $\mathbf{V}$ computes

$$r = e(U,P)e\Big(\sum_{i=1}^{n} Q_i, -P_{pub}\Big)^c.$$

Accepts the multisignature if and only if $c = h(m,r)$.

4.2. *Correctness*

The verification process of the multisignature $\langle U, c\rangle$ on message m is acceptable as given by the following equations:

$$\begin{aligned}
&e(U,P)e\Big(\sum_{i=1}^{n} Q_i, -P_{pub}\Big)^c\\
&\quad = e\Big(\sum_{i=1}^{n} U_i, P\Big)e\Big(\sum_{i=1}^{n} Q_i, -P_{pub}\Big)^c\\
&\quad = e\Big(\sum_{i=1}^{n}(cS_{ID_i} + k_iP), P\Big)e\Big(\sum_{i=1}^{n} Q_i, -P_{pub}\Big)^c\\
&\quad = e\Big(\sum_{i=1}^{n}(cS_{ID_i}, P\Big)e\Big(\sum_{i=1}^{n} k_iP, P\Big)e\Big(\sum_{i=1}^{n} Q_i, -P_{pub}\Big)^c\\
&\quad = e\Big(\sum_{i=1}^{n} Q_i, P_{pub}\Big)^c\prod_{i=1}^{n} e(k_iP, P)e\Big(\sum_{i=1}^{n} Q_i, -P_{pub}\Big)^c\\
&\quad = \prod_{i=1}^{n} e(k_iP, P) = \prod_{i=1}^{n} r_i = r.
\end{aligned}$$

Thus the correctness of the scheme is justified.

5. Security Analysis

The notion of security for a multisignature has to capture the possibility of an adversary to “forge” a set $\mathbf{S}$ of signers and a multisignature of some message such that the latter is accepted by a verifier when not all signers of the set did sign the message. In other words, no valid multisignature should keep an honest signer, who is part of the set $\mathbf{S}$, accountable if he did not participate in signing.

In order to achieve its goal an adversary might corrupt signers and send arbitrary messages during multisignature generation, etc. We allow an adversary to extract private keys of arbitrary entities for identities of their choice. We also allow an adversary to corrupt all but one player and its goal is to “frame” the honest player S_h. We now formalize the notion of security for identity based multisignatures. It is similar to the one given in (Micali *et al.*, 2001; Boldyreva, 2003).

DEFINITION. An adversary A learns the system parameters `params` and identity ID_h of the single honest signer. A extracts private keys corresponding to the rest $(n-1)$ signers' identities. A is allowed to run multisignature generation with the honest player on behalf of $(n-1)$ corrupted signers on the chosen message m'. The advantage of adversary $Adv(A)$ is defined as the probability of A to output the valid message-set-signature triple $(m',\mathbf{S}, Sig)$, such that $S_h \in \mathbf{S}$, $MV(m',\mathbf{S},Sig) = 1$ and S_h did not participate in the multisignature generation on the input message m'.

We say that a multisignature scheme MS is secure against existential forgery under chosen message attack if there exists a polynomial-time adversary A with non-negligible advantage $Adv(A)$.

In Theorem 1 of (Hess, 2003), Hess proved that his ID-based signature scheme is secure against existential forgery under adaptive chosen message attack in the random oracle model.

Theorem. *The SMS and PMS are secure serial and parallel multisignature schemes in the random oracle model.*

Proof. Let A_{SMS} and A_{PMS} be polynomial-time adversaries for our SMS and PMS respectively. Let A_{HS} be a polynomial-time adversary for our base scheme (Hess, 2003). Utilizing the result of Theorem 1 of (Hess, 2003), we prove that our SMS and PMS are secure against existential forgery under chosen message attack in the random oracle model.

The idea behind this proof is that if A_{SMS} or A_{PMS} manages to frame an honest signer by constructing a valid multisignature on an arbitrary message without interacting with this honest signer, then A_{HS} can forge a previously unsigned message. A_{HS} can query the hash and signing oracles with an identity ID for any arbitrary message. Whenever A_{SMS} or A_{PMS} wants to get a valid multisignature by framing an honest signer, it sends the respective signing query to A_{HS}. A_{HS} forwards the signing query given by A_{SMS} to its signing oracle with the identity and the partial multisignature as input. A_{PMS} sends the message and identity of the honest user to the A_{HS}, which it forwards to its signing oracles. A_{HS} returns the respective reply of signing oracle to A_{SMS} or A_{PMS}. It is easy to see that A_{SMS} or A_{PMS} will be successful in its attempts if the reply from A_{HS} is a valid signature. But, A_{HS} generating a valid signature is a clear contradiction to the result of the Theorem 1 of (Hess, 2003). Hence the proof.

6. Conclusion

We proposed identity based serial and parallel multisignature schemes using bilinear pairings. Our serial multisignature scheme logically requires a forced verification at each level, avoiding the overlooking in verifying the signature of the predecessor. We also presented a parallel multisignature scheme in which the verification of individual signatures is performed by a designated clerk. We proved that the schemes are secure against existential forgery under adaptive chosen message attack in the random oracle model.

References

Bellare, M., and P. Rogaway (1993). Random oracles are practicle – a paradigm for designing efficient protocols. In *First ACM Conference and Communications Security*. ACM. pp. 62–73.

Boldyreva, A. (2003). Threshold signatures, multi signatures and blind signatures based on the GDH group signature scheme. In *PKC 2003*, *Lecture Notes in Computer Science*, vol. 2567. Springer-Verlag. pp. 31–46.

Boneh, D., and M. Franklin (2001). Identity based encryption from the weil pairing. *SIAM Journal of Computing*, **32**(3), 586–615. Extended abstract in *Proceedings of* CRYPTO 2001, *Lecture Notes in Computer Science*, vol. 2139. Springer-Verlag. pp. 213–229.

Boneh, D., B. Lynn and H. Shacham (2002). Short signatures from the Weil pairing. In *Asiacrypt 2001*, *Lecture Notes in Computer Science*, vol. 2248. Springer-Verlag. pp. 514–532.

Boyd, C. (1989). Digital multisignatures. In *Cryptography and Coding*. Oxford University Press. pp. 241–246.

Chen, X., F. Zhang and K. Kim (2003). ID-based multi-proxy signature and blind multisignature from bilinear pairings. In *Proceedings of* KIISC'2003. pp. 11–19.

Gangishetti, R., M.C. Gorantla, M.L. Das, A. Saxena and V.P. Gulati (2005). An efficient secure key issuing protocol in ID-based cryptosystems. In *Proceedings of the International Conference on Information Technology*: *Coding and Computing* (*ITCC 2005*), vol. 1. IEEE Computer Society. pp. 674–678.

Gorantla, M.C., R. Gangishetti and A. Saxena (2005). A survey on ID-based cryptographic primitives. In *IACR Cryptology ePrint Archive*. Report 2005/094.
Available at `http://eprint.iacr.org/2005/094.`

Guttman, P. (2002). PKI: Its not dead, just resting. *IEEE Computer*, **35**(8), 41–49.
Extended version available at `www.cs.auckland.ac.nz/ pgut001/pubs/notdead.pdf.`

Harn, L. (1994). Group-oriented (t,n) threshold digital signature scheme and multisignature. In *IEE Proceedings – Computers and Digital Techniques*, **141**(5), 307–313.

Hess, F. (2002). An efficient identity based signature schemes based on pairings. In *SAC 2002*, *Lecture Notes in Computer Science*, vol. 2595. Springer-Verlag. pp. 310–324.

Horster, P., M. Michels and H. Petersen (1995). Meta-multisignatures schemes based on the discrete logarithm problem. In *IFIP/Sec 1995*. pp. 128–142.

Itakura, K., and K. Nakamura (1983). A public key cryptosystem suitable for digital multi signatures. *NEC Research and Development*, **71**, 1–8.

Joux, A. (2000). A one round protocol for tripartite Diffie–Hellman. In *Algorithmic Number Theory, 4th International Symposium, ANTS-IV*. LNCS 1838, Springer. pp. 385–394.

Lin, C.Y., T.C. Wu and J. Hwang (2001). ID-based structured multi signature schemes. In *Advances in Network and Distributed Systems Security* (*IFIP Conference Proceedings 206*). Kluwer Academic publishers. pp. 45–59.

Micali, S., K. Ohta and L. Reyzin (2001). Accountable subgroup multi signatures. In *ACM Conference on Computer and Communications Security*, *ACM*. pp. 245–254.

Mitchell, C.J. (2001). An attack an ID-based nulti signature sheme. In *Royal Holloway, University of London, Mathematics Department Technical Report*. RHUL-MA-2001-9.

Ohata, T., and T. Okamoto (1991). A digital multisignature scheme based on the Fiat–Shamir scheme. In *Asiacrypt 91*, *Lecture Notes in Computer Science*, vol. 739. Springer-Verlag. pp. 75–79.

Ohata, K., and T. Okamoto (1999). Multi signature scheme secure against active insider attacks. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, **E82-A**(1), 21–31.

Okamoto, T. (1988). A digital multi signature schema using bijective public key crypto systems. *ACM Transactions on Computer Systems*, ACM, **6**(4), 432–441.

Shamir, A. (1985). ID-based cryptosystems and signature schemes. In *Proceedings of* Crypto 84, *Lecture Notes in Computer Science*, vol. 196. Springer. pp. 47–53.

R. Gangishetti completed his master's degree of technology in IT with specialization in banking technology and information security from University of Hyderabad. His research interests include cryptography, information and network security.

M.C. Gorantla completed his master of technology in IT with specialization in banking technology and information security from University of Hyderabad. His research interests include cryptography, network security and data structures.

M.L. Das received his m. tech. degree in 1998. He is currently pursuing his PhD work in K.R. School of Information Technology, Indian Institute of Technology – Bombay, India. He is a member of Cryptology Research Society of India and Indian Society for Technical Education. His research interests include cryptography and information security.

A. Saxena received his PhD degree in computer science. He is an associate professor with Institute for Development and Research in Banking Technology, Hyderabad, India. He is member IEEE and life member of Cryptology Research Society of India and Computer Society of India. His research interests include authentication technologies, smart cards, key management and security issues in banking.

Identiškumu grindžiamos daugelio parašų schemos

Raju GANGISHETTI, M. Choudary GORANTLA, Manik Lal DAS, Ashutosh SAXENA

Straipsnyje pateikiamos nuoseklios ir lygiagrečios identiškumu grindžamos daugelio parašų schemos, naudojančios bitiesinius poravimus. Kiekviename nuosekliosios daugelio parašų schemos lygyje atliekamas priverstinis verifikavimas kad išvengus ankstesnio lygio parašo atskleidimo. Tuo tarpu lygiagrečioje daugelio parašų schemoje individualiu parašu verifikavimą atlieka paskirtas tarnautojas. Straipsnyje parodyta, kad pateiktos shemos yra saugios nuo egzistencinių klastojimų naudojant adaptyvaus pranešimo ataką atsitiktiniame oraklo modelyje.