Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
October 2012, 19(Suppl. 2): 122–127
www.sciencedirect.com/science/journal/10058885
The Journal of China Universities of Posts and Telecommunications
http://jcupt.xsw.bupt.cn

Novel self-renewal Hash chain based on Ito-Saito-Nishizeki secret sharing scheme

DAI Ting[1,2], HUANG Hai-ping[1,2] (✉), WANG Ru-chuan[1,2,3], PAN Xin-xing[1]

1. College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
2. Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China
3. Key Laboratory of Broadband Wireless Communication and Sensor Network Technology of Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

Abstract

Hash chain mechanism has been widely used into a variety of encryption applications and services. The introduction of renewable Hash chain overcomes resource-constrained defect in traditional Hash chains, but many current renewable schemes still hold unsatisfactory performance especially on security and complexity. This paper proposed a novel self-renewable Hash chain construction scheme based on Ito-Saito-Nishizeki secret sharing scheme (SSS). It has proved that the proposed Hash chain scheme has higher security and less consumption of communication, computation and memory than the typical schemes proposed in the existing literatures.

Keywords security, renewable Hash chain, Ito-Saito-Nishizeki SSS

1 Introduction

Hash function is characterized by one-wayness and high computational efficiency, therefore, Hash chain mechanism has been widely used in various encryption applications and services, such as one-time password system [1], digital signature mechanism, key distribution scheme [2], micropayment protocol [3], broadcast authentication [4], video streaming security [5–6] and so on. However, most of these applications are constrained by the resource of Hash chain with finite length [7]. It is a strange phenomenon that the length of Hash chain contradicts itself. On one hand, with a short length, Hash chain will be consumed quickly. When it is exhausted, the system needs to be reinitialized to generate a new chain in virtue of public key signature technology, which seriously reduces the efficiency and increases the computational complexity. On the other hand, with a long length, Hash chain will increase the storage consumption immediately. The longer the chain is the lower service efficiency it will have. Because the length of the chain is the number of Hash calculations the initialization needs to execute, the longer one will increase burden and make time assignment unreasonable.

To solve this contradiction, many papers have proposed new Hash chain schemes. In 2004, Goyal proposed the re-initializable Hash chain (RHC) scheme (http://eprint.iacr.org/2004/097.pdf), whose main idea is that when a RHC is exhausted, a new RHC can be regenerated safely and undeniably.

In 2006, on the basis of RHC, Ref. [7] proposed the elegant re-initializable Hash chain (ERHC) scheme. In the initialization phase, the sender generates $\left(L+\lfloor \mathrm{lb}(L) \rfloor+1\right)$ random numbers, and unites them into S_U, and hashes them, and unites the Hash results into P_U. Then, treat P_U as seed to compute a Hash chain with the tip $h^k(P_U)$, which would be sent securely. And when a Hash chain is exhausted, the sender generates new instances S'_U, P'_U and a new chain with tip $h^k(P'_U)$. As for $h^k(P'_U)$, part of the corresponding S_U needs to be sent. And then the

Received date: 03-07-2012
Corresponding author: HUANG Hai-ping, E-mail: hhp@njupt.edu.cn
DOI: 10.1016/S1005-8885(11)60433-0

receiver can verify it by comparing P_U with the received part of S_U. ERHC smoothly generates a new chain when the old one is exhausted, so, it constructs infinite length Hash chain logically. However, it may be chosen-plaintext attacked owning to publishing part of S_U to achieve verification.

In 2008, Ref. [8] proposed the self-updating Hash chain (SUHC) scheme based on hard core predicate algorithm. The main idea of SUHC is that it distributes the first chain's every key value with one bit of the second one's tip. In this manner, the receiver would gain all bits of second chain's tip when the first one is exhausted.

In the same year, on the basis of Ref. [8], Ref. [9] proposed the self-renewal Hash chain (SRHC) scheme. The primary distinction between the above two schemes is the different selection algorithm of random numbers. SUHC selects a random number named SR_i meeting the condition of $B(\mathrm{SR}_i)=\omega'[i]$, and obtains PR_i by hashing SR_i. While SRHC selects a random number R_i meeting the condition of $B(h^{k-i}(s)||R_i)=\omega'[i]$. The security distributions of tips of SUHC and SRHC all rely on the security distribution of k random numbers; k denotes the length of chain. And the above two schemes demand all the random numbers received integrally and inevitably. Thus, the tip of the new chain can be reconstructed. Obviously, they have abandoned the original fault-tolerance of Hash chain.

The analyses of the above typical schemes have demonstrated that whether RHC, ERHC or SUHC, SRHC, they all transform or map every bit of the new chain's tip into a random number, making the security of the new tip decided by the safe distribution of the random numbers. In addition, only if they receive all the random numbers integrally would they regenerate the new tip correctly. Consequently, they all weaken the security and increase the consumptions.

Aimed at overcoming the inadequacies of the above schemes, this paper proposes a novel self-renewal Hash chain scheme based on Ito-Saito-Nishizeki SSS (SRHC-ISN). The rest of this paper is organized as follows. Sect. 2 introduces some new definitions and conceptions. Sect. 3 presents the whole process of SRHC-ISN elaborately. Sect. 4 analyses the performances of SRHC-ISN and presents the comparisons of SRHC-ISN with RHC, ERHC, SUHC and SRHC. Sect. 5 concludes the paper.

2 Background knowledge

In this section, we intend to introduce some basic facts about SSS and the definition about Ito-Saito-Nishizeki SSS. Then, we briefly present the process of integrating $\boldsymbol{C}^{\mathrm{A}}$ with Hash chain's seed s.

2.1 Basic facts about SSS

Let n be an integer, $n\geqslant 2$, the set of participants be $P=\{p_1,p_2,...,p_n\}$, and an access structure A defined on P comprises of a collection of subsets of P. A is a monotone access structure if whenever $a\in \mathrm{A}$ and $b\supseteq \mathrm{A}$, then $b\in a(a,b\subseteq P)$. And $\overline{\mathrm{A}}=P(\{1,2,...,n\})/\mathrm{A}$ is called the unauthorized access structure.

Definition 1 The sets $C\in \mathrm{A}$ with $C/c\notin \mathrm{A}$ for all $c\in C$ are called the minimal sets of A, the collection of which is denoted by $\mathrm{A}_{\min}$. The sets $C\in \overline{\mathrm{A}}$ with $C\bigcup c\in \mathrm{A}$ for all $c\in \overline{C}=P/C$ are called the maximal unauthorized sets of $\overline{\mathrm{A}}$, the collection of which is denoted by $\overline{\mathrm{A}}_{\max}$.

Equivalently, an A-SSS is a method of generating (S, (I_1,…, I_n)) such that

1) for any $a\in \mathrm{A}$, finding the element S, given the set $\{I_i \mid i\in a\}$ is easy;

2) for any $a\in \overline{\mathrm{A}}$, finding the element S, given the set $\{I_i \mid i\in a\}$ is intractable;

The set A will be referred to as the authorized access structure or simply as the access structure, S will be referred to as the secret and I_1, ..., I_n will be referred to as the shares (or the shadows) of S. The elements of the set A will be referred to as the authorized access sets of the scheme.

Definition 2 A cumulative map (α, S) for A is a finite set S accompanied by a mapping $\alpha: P\rightarrow 2^S$ (where 2^S is the collection of all subsets of S) such that for $Q\subseteq P$, $\bigcup_{a\in Q} a^{\alpha}=S \Leftrightarrow Q\in \mathrm{A}$.

2.2 Ito-Saito-Nishizeki SSS

Definition 3 Let A be a monotone authorized access structure of size n and let B_1, …, B_m be the corresponding maximal unauthorized access sets. The cumulative array for the access structure A, denoted by $\boldsymbol{C}^{\mathrm{A}}$, is the $n\times m$ matrix, where $(\boldsymbol{C}^{A}_{i,j})_{1\leqslant i\leqslant n,\ 1\leqslant j\leqslant m}$ where

$$\boldsymbol{C}^{\mathrm{A}}_{i,j}=\begin{cases}0, & \text{if } i\in B_j\\ 1, & \text{if } i\notin B_j\end{cases}$$

for all $1\leqslant i\leqslant n$ and $1\leqslant j\leqslant m$.

Let consider now an arbitrary (*m*, *m*)-threshold SSS with the secret *S* and the corresponding shadows s_1,..., s_m. In the A-SSS, the shadows I_1,..., I_n corresponding to the secret *S* will be defined as

$$I_i = \left\{ s_j \middle| \boldsymbol{C}^{\mathrm{A}}_{i,j}=1 \right\}$$

for all $1 \leqslant i \leqslant n$.

Based on Definition 3, choose a specified integer *n* and a specified $\mathrm{A}_{\min}$. In such a way, it is easy to obtain the corresponding $\overline{\mathrm{A}}_{\max}$ and *m*. Then, the cumulative array $\boldsymbol{C}^{\mathrm{A}}$ for the access structure A can be computed simply. Since $\boldsymbol{C}^{\mathrm{A}}$ is an $n \times m$ matrix, unfold it by row to form a row matrix with $n \times m$ columns, which can be viewed as a binary number with length of $n \times m$, denoted by *s*.

3 Hash chain scheme

In this section, we propose to design a new construction scheme of re-initializable Hash chain based on Ito-Saito-Nishizeki SSS. In the scheme, there are two entities: a distributor and an authenticator. The former one generates a Hash chain and employs link values of Hash chain as its authentication proofs or tokens, while the later one authenticates the validity of these proofs.

3.1 Initialization

In initial phase, the distributor and the authenticator negotiate the length of Hash chain *n*, a secure Hash function $h: \{0,1\}^* \to \{0,1\}^k$ with a security parameter *k*, which means the output of *h* is a *k*-bits string. And they also negotiate the initial time T_0 and the time threshold T_{int}, which stipulates the valid time interval of each link value.

1) Initialize a Hash chain. Based on Ito-Saito-Nishizeki SSS, compute the initial seed *s* and generate a Hash chain of length *n*.

$s, h(s), h^2(s),..., h^n(s)$

2) Generate the next chain. The certifier choose a new *n* and $\mathrm{A}_{\min}$, computer the new seed s', and generate a new Hash chain.

$s', h(s'), h^2(s'),..., h^n(s')$

3) Regard $h^n(s')$ as the secret *S*, divide it into *m* parts, denoted by $s_1,\ldots, s_m$. Then, on basis of matrix $\boldsymbol{C}^{\mathrm{A}}$, computer *n* shadows $I_1,\ldots, I_n$.

4) Compute the message authentication code (MAC), denoted by $\xi_0 = h^{n-1}(s) \oplus I_1$.

5) Publish $h^n(s)$ to the authenticator securely.

3.2 Distribution

In the phase of distribution, the distributor computes and distributes link values and tokens for authentication. For the *i*th (i=1, 2,…, $n-1$) distribution the distributor does:

1) Compute the link value of Hash chain $h^{n-i-1}(s)$ and $h^{n-i}(s) = h(h^{n-i-1}(s))$.

2) Compute the MAC, denoted by $\xi_i = h^{n-i-1}(s) \oplus I_{i+1}$.

3) Construct and distribute the certification frame ($h^{n-i}(s)$, I_i, ξ_i).

And in the *n*th distribution, publish the secret (s, I_n).

3.3 Authentication

For the *i*th authentication the verifier does:

(1) If the receiving time is less than or equal to $T_0+i\times T_{\mathrm{int}}$, then receive the certification frame ($h^{n-i}(s)$, I_i, ξ_i) from the distributor.

1) Compute and verify whether $h(h^{n-i}(s))$ equals to $h^{n-i+1}(s)$, where $h^{n-i+1}(s)$ is a link value sent in the last valid session and saved.

2) Compute and verify whether MAC $h^{n-i}(s) \oplus I_i$ equals to ξ_{i-1}.

If all checks are passed, then the authenticator verifies the distributor successfully and then stores the shadows I_i.

(2) If the receiving time is more than $T_0+i\times T_{\mathrm{int}}$, drop $h^{n-i}(s)$ and I_i, save ξ_i. Then, it will wait for the next valid certification frame ($h^{n-j}(s)$, I_j, ξ_j), where $j > i$.

1) Compute and verify whether $h^{j-i+1}(h^{n-j}(s))$ equals to $h^{n-i+1}(s)$, where $h^{n-i+1}(s)$ is a link value sent in the last valid session and saved.

2) Compute and verify whether MAC $h^{n-j}(s) \oplus I_j$ equals to ξ_{j-1}.

If all checks are passed, then the authenticator verifies the distributor successfully and then stores the shadows I_i.

3.4 Recombination

After all link values been distributed, that is to say, the whole Hash chain exhausted, the authenticator has stored the seed *s* and *n* or fewer shadows I_i.

Compute the cumulative array $\boldsymbol{C}^{\mathrm{A}}$. In a contrary manner to constructing *s*, the authenticator can regenerate $\boldsymbol{C}^{\mathrm{A}}$ from the known *s*, *n*.

Compute the secret *S*. Based on $\boldsymbol{C}^{\mathrm{A}}$, pick s_i form I_j in

orders, where $i=1,2,...,m$ and $j=1,2,...,n$. And it is easy to recombine S from s_1, s_2,..., s_m. That means the authenticator obtains the next chain's tip $h^n(s')$.

3.5 Self-renewal

After recombining, the next chain starts to work and another new chain is generated including a seed s'' and the corresponding tip $h^n(s'')$. Iteration of above processes has a result that Hash chains work continuously and infinitely.

4 Analysis

In this section, we intend to discuss security, validity and complexity of the proposed Hash scheme in Sect. 3.

4.1 Security

The security of this scheme is based on one-way function and Ito-Saito-Nishizeki SSS. In the phase of distribution, the purpose of XORing Hash link value $h^{n-i-1}(s)$ and the shadow I_{i+1} is to maintain the integrity and confidentiality of I_{i+1}, while the adoption of delaying the disclosure of I_i in certification frame is to guarantee the non-repudiation.

Meanwhile, in the phase of authentication, the tolerance of message loss or fault is embodied in the scheme. Because, in Ito-Saito-Nishizeki SSS, $\boldsymbol{C}^{\mathrm{A}}$ is an $n\times m$ matrix, which means that every secret shadow s_i averagely appears $n/2$ times in the n shadows I_1, I_2,…, I_n for all $1\leqslant i\leqslant m$. Thus, even the worst, there are more than $n/2$ certification frames lost or dropped, part of the secret S can also be verified by the less than m valid s_i for all $1\leqslant i\leqslant m$. Not mention that, under normal circumstances, with the low packet loss or fault ratio, our scheme can work steadily and strongly.

Twice authentication processes in the phase of authentication have been proposed to strengthen the security and integrity. The first process occurs at the time when the authenticator judges whether the receiving time is valid. Only the $h^{n-i}(s)$ and I_i in valid time interval are stored to be authenticated twice. And the second process happens when the authenticator judges whether $h^{n-i}(s)$ is valid through the authenticated and stored $h^{n-i+1}(s)$, and whether I_i is valid through $h^{n-i}(s)$ and ξ_{i-1}. Only passing the twice authentications can the disclosure packet of shadow I_i be accepted.

In addition, the sequence of s_i is based on the seed s, which is one of the critical factors of this scheme. And the sequence is exactly the matrix $\boldsymbol{C}^{\mathrm{A}}$, which is also the seed s of the first Hash chain. Thanks to one-wayness of Hash function again, before distributing s, all the secret shadows s_i cannot be differentiated from the shadows I_1, I_2,…, I_n, not mention to put them in order.

4.2 Validity

The proposed scheme distributes the Hash link value and the deformed shadow information first, and then publishes the link value with the shadow. Because of the non-repudiation and one-wayness of Hash function, the scheme can correctly authenticate the shadows of secret S. And with the matrix $\boldsymbol{C}^{\mathrm{A}}$ computed by s, it can put the secret shadows s_i in order to reconstruct the secret S, which is just the tip $h^n(s')$ of the next chain.

4.3 Complexity

The consumptions of SRHC-ISN have been compared with the ones of RHC, ERHC, SUHC and SRHC as follows, from three aspects, computation, communication and memory. Table 1 shows the detailed parameters of the above schemes.

Here are some explanations about the mentioned parameters.

1) k means the output of Hash function is a k-bits string.

2) n means the length of Hash chain.

3) m means the number of secret shadows in SRHC-ISN.

4) H means the computation consumption of the Hash function.

5) U means the computation consumption of the union operation.

6) R, R_B, $R_{B'}$ means the computation consumptions of generating a random number in RHC, ERHC, SUHC or SRHC respectively.

7) B, B' means the computation consumption of obtaining one bit from a random number by hard core predicate in SUHC and SRHC respectively.

8) C_{A}, I, P means the computation consumption of generating a matrix $\boldsymbol{C}^{\mathrm{A}}$, computing the shadows I_j and picking secret shadows s_i form I_j in SRHC-ISN respectively.

9) X means the computation consumption of XOR.

10) len_{H} means the communication or memory

consumption of k (bit).

11) len_s means the communication or memory consumption of the seed of Hash chain.

12) len_r means the communication or memory consumption of the generated random number.

13) len_I means the communication or memory consumption of shadows I_j in SRHC-ISN.

14) len_{se} means the communication or memory consumption of the secret shadows s_i in SRHC-ISN.

Table 1 The consumptions in RHC, ERHC, SUHC, SRHC and SRHC-ISN

RHC	Initialization	Comp.	$(3k+2)\cdot H+2R$
		Comm.	2len_H
		Memo.	$2(\text{len}_s+\text{len}_r)+4\text{len}_H$
	Distribution Authentication Combination	Comp.	$(1/2)(k^2+3k-4)H+2(k-1)R$
		Comm.	$3k\text{len}_r+6k-2$
		Memo.	$\text{len}_r+(k+2)\text{len}_H+k$
ERHC	Initialization	Comp.	$2(n+k+\lfloor \text{lb}k \rfloor+1)\cdot H+2(k+\lfloor \text{lb}k \rfloor+1)\cdot R+2U$
		Comm.	0
		Memo.	$(2k+1)\text{len}_H+2k\text{len}_r$
	Distribution Authentication Combination	Comp.	$(1/2)(n^2+n+k+\lfloor \text{lb}k \rfloor+1)H$
		Comm.	$2(n+k+\lfloor \text{lb}k \rfloor+1)\text{len}_H+(k+\lfloor \text{lb}k \rfloor+1)\text{len}_r$
		Memo.	$(n+k+\lfloor \text{lb}k \rfloor+1)\text{len}_H+k$
SUHC	Initialization	Comp.	$(3k+1)\cdot H+R_B$
		Comm.	2len_H
		Memo.	$2\text{len}_s+\text{len}_r+4\text{len}_H$
	Distribution Authentication Combination	Comp.	$(1/2)(k^2+6k-4)H+(k-1)R_B+kB$
		Comm.	$(6k-3)\text{len}_H+2k\text{len}_r$
		Memo.	$(k+2)\text{len}_H+\text{len}_r+k$
SRHC	Initialization	Comp.	$3kH+R_{B'}$
		Comm.	2len_H
		Memo.	$2\text{len}_s+\text{len}_r+3\text{len}_H$
	Distribution Authentication Combination	Comp.	$(1/2)(k^2+5k-2)H+(k-1)R_{B'}+kB'$
		Comm.	$(4k-2)\text{len}_H+2k\text{len}_r$
		Memo.	$k\text{len}_H+2\text{len}_r+k$
SRHC-ISN	Initialization	Comp.	$(3n-1)H+2C_A+nI+X$
		Comm.	0
		Memo.	3len_H+2k
	Distribution Authentication Combination	Comp.	$(1/2)(n^2+n)H+(2n-1)X+nP+U$
		Comm.	$(4n-2)\text{len}_H+2n\text{len}_I$
		Memo.	$n\text{len}_H+n\text{len}_I+m\text{len}_{se}$

Briefly, suppose that $k\approx n$, $R\approx R_B\approx R_B$, $B\approx B'$, $H>U$, $H>C_A>I>P$, $\text{len}_H \gg \text{len}_s \approx \text{len}_r \approx \text{len}_I > \text{len}_{se}$, then it is easy to compare the consumptions of SRHC-ISN with RHC, ERHC, SUHC, SRHC, as shown in Table 2.

Table 2 The comparisons of RHC, ERHC, SUHC, SRHC and SRHC-ISN

Initialization	Comp.	SRHC-ISN<SRHC<SUHC<RHC<ERHC
	Comm.	ERHC=SRHC-ISN<RHC=SRHC=SUHC
	Memo.	SRHC-ISN<SRHC<SUHC<RHC<ERHC
Distribution Authentication Combination	Comp.	SRHC-ISN<ERHC<RHC<SRHC<SUHC
	Comm.	SRHC<SRHC-ISN<ERHC<RHC<SUHC
	Memo.	SRHC<SRHC-ISN<SUHC=RHC<ERHC

The comparisons in Table 2 show that the proposed scheme has great performance in the complexity. The consumptions of SRHC-ISN in the initialization phase are much less than other schemes. While in the phase of distribution-authentication-combination, SRHC-ISN's consumptions of communication and memory are a little more than SRHC's but much less than RHC's, ERHC's and SUHC's. In addition, the computation consumption of SRHC-ISN is always the least.

5 Conclusions

Renewable Hash chains can overcome resource-constrained defect in traditional ones. In recent years, many literatures have proposed the different and new construction schemes of renewable Hash chains, like RHC, ERHC, SUHC and SRHC. However, the above four schemes have security deficiency or consumption deficiency more or less. Aimed at their inadequacies, based on Ito-Saito-Nishizeki SSS, this paper has proposed a novel self-renewable Hash chain scheme, SRHC-ISN. It has theoretically proved that SRHC-ISN maintains integrity, confidentiality and non-repudiation by the adoption of delaying disclosure and one-wayness. In addition, it can also tolerate message loss or fault by the property of the shadows in Ito-Saito-Nishizeki SSS. Furthermore, the use of twice authentication and transformed secret shadows guarantees the scheme stronger security. Besides, the analysis of complexity has shown that SRHC-ISN has a less consumption than the proposed typical schemes in the mass.

Acknowledgements

The work was supported by the National Natural Science Foundation of China (60973139, 61170065, 61003039), the Natural Science Foundation of Jiangsu Province(BK2011755), the Scientific and Technological Support Project (Industry) of Jiangsu Province (BE2010197, BE2010198, BE2012183), the Natural Science Key Fund for Colleges and Universities in Jiangsu Province (11KJA520001, 12KJA520002), the Project Sponsored by Jiangsu Provincial Research Scheme of Natural Science for Higher Education Institutions (10KJB520013, 11KJB520014, 11KJB520016), the Scientific Research and Industry Promotion Project for Higher Education Institutions (JH2010-14, JHB2011-9), the Postdoctoral Foundation (2012M511753, 1101011B), the Science and Technology Innovation Fund for Higher Education Institutions of Jiangsu Province (CXZZ11-0405, CXZZ11-0406), the Doctoral Fund of Ministry of Education of China (20103223120007, 20113223110002), the Key Laboratory Foundation of Information Technology Processing of Jiangsu Province (KJS1022), the Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (yx002001).

References

1. Mohamed H E, Muhammad K K, Khaled A. One-time password system with infinite nested Hash chains. Communications in Computer and Information Science, 2010, 122: 161–170
2. Ramkumar M, Memon N. An efficient key pre-distribution scheme for MANET Security. IEEE Journal on Selected Areas of Communication, Mar 2005
3. Chen L, Zhang H J, Liu N. Authentication and micropayment protocols based on self-updating Hash chains. The Sixth International Conference on Grid and Cooperative Computing (GCC 2007)
4. Liu D G, Ning P. Multi-level μTESLA: broadcast authentication for distributed sensor networks. ACM Transaction on Embedded Computing System (TECS), 2004, 3(4): 800–836
5. Emad A E, Mohammed B, Hossam A. Hash chain links resynchronization methods in video streaming security: performance comparison. Mobile Multimedia, 2011, 7(1): 89–112
6. Gabriele O, Stefano C, Roberto D P, et al. Robust and efficient authentication of video stream broadcasting. ACM Transactions on Information and System Security (TISSEC), 2011, 14(1): 1–25
7. Zhao Y C, Li D B. An elegant construction of re-initializable Hash chains. Electronics & Information Technology, 2006, 28(9): 1717–1720 (in Chinese)
8. Zhang H J, Zhu Y F. Self-updating Hash chains and their implementations. Lecture Notes in Computer Science, 2006, 4255: 387–397
9. Zhang H J. A novel self-renewal Hash chain and its implementation. IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC'08), Shanghai, 2008, 2: 144–149

From p. 46

13. Masazade E, Rajagopalan R, Varshney P K, et al. A multiobjective optimization approach to obtain decision thresholds for distributed detection in wireless sensor networks. IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics, 2010, 40: 444–457
14. Ermis E B, SaligramaV. Distributed detection in sensor networks with limited range multimodal sensors. IEEE Transactions on Signal Processing, 2010, 58: 843–858
15. Aziz A. A soft-decision fusion approach for multiple-sensor distributed binary detection systems. IEEE Transactions on Aerospace and Electronic Systems, 2011, 47: 2208–2216