Intelligence Analysis and Processing System for Semantic Concept Network Based-on State Identification

Ruo Hu
Department of Computer Science, GuangDong Polytechnic Normal University
GuangZhou city, China, Email: hu68@163.com

***Abstract*—In this paper, we present an intelligence analysis and processing system based on state classify for the Semantic Concept Network (SCN). We argue that due significant change in costs of computation and resource distribution, our system is particularly suitable for the SCN environment with important resources. One of the interesting natures of our system is that it provides time-sharing distribution of the offline resource, which allows the user to reuse the offline pre-computed information in assembly time, in contrast to one-time distribution in all previous Intelligence Analysis and Processing (IAP) classify systems. As evidence of the feasibility and practicality of our system to be used in the SCN environment, we provide an real implementation result of our system on the Micro-V platform.**

***Index Terms*—IAP, Semantic Concept Network, time-sharing, resource**

I. Introduction

SCN Applications and Security. A Semantic Concept Network (SCN).is a concept network consisting of spatially distributed autonomous devices using concept to cooperatively represent physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. There are many potential applications for SCN [1]. They could be used in commercial and industrial applications to monitor data that would be difficult or expensive to monitor using wired network. They could be used to monitor situations in some hazard environments, such as in nuclear power plants. They could also be deployed in wilderness areas, where they would remain in operation for many years (monitoring some environmental variables) without the need to recharge/replace their power supplies. They could form a perimeter about a property and monitor intruders.

SCN are more vulnerable to various attacks due to their nature of wireless communication. In some SCN applications, providing authentication for sensed data is of prime importance. For example [6], in radiological facilities where node collect data on radioactive levels of nuclear power plants and transmit them to base stations or workers' dosimeters, it should be assured that the collected data are authentic and have not been altered during transmission in order to avoid malfunction or other possible hazards to the workers due to misinterpretation caused by altered data. State-based cryptography could particularly be suitable for SCN. The absence of certificate eliminates the costly certificate verification process. In addition, when there is a new node added to the network, other nodes do not need to have its certificate in order to communicate in a secure and authenticated way. This can greatly reduce communication overhead and computation cost, which is a significant factor in the design of SCN. Recently, Tan et al. [5] proposed a state-based encryption method for body node network (BNN), a network of node deployed on a person's body to collect physiological information.

Intelligence analysis and processing Classify. In order to further reduce the computational cost of classify generation, intelligence analysis and processing classify is preferable in SCN. The notion of intelligence analysis and processing classifys was introduced by expert [6]. It performs the classify generation procedure in two phases. The first phase is performed offline (prior to the knowledge of the message to be signed), and the second phase is performed online (after knowing the message to be signed). In SCN, the offline phase can be executed at the base station, while the online phase is to be executed in the SCN node. The online phase is typically very fast and hence can be executed efficiently even on a weak processor, such as a node in SCN.

A. Related Works

The only existing classify-based intelligence analysis and processing classify method was designed by Xu, Mu and Wang (this method will be referred to as the "XMS" method hereafter). In their method, the user needs to execute the offline phase every time when he wants to produce a classify .We call it "one-time" meaning the offline classify part can be used only once, and hence, it cannot be re-used. If we apply this one-time method into SCN, it becomes impractical since, assuming the offline phase is done at the base station, non-reusability of the resource distribution implies that Semantic Concept need to go back to the base station every time for obtaining the

Manuscript received March. 1, 2012; revised April 1, 2012; accepted May 20, 2012.
Project number: 10771039

doi:10.4304/jcp.8.5.1315-1320

next offline classify part. Moreover, the verification of the XMS method requires a pairing operation, which is a costly computation process for a node. We do not expect a node can execute such a heavy operation, which makes the classify method not appropriable for node-to-node classifys in SCNs.1

B. The Current Challenges of Semantic Concept Networks

The current challenges in Semantic Concept Networks are: to ensure an efficient and full use of Semantic Concept resources and multimedia applications, Connect at best anywhere, anytime and with any network. Customize the more powerful natures stimulated by the increasing consumers' demand. Find solutions for the mobile business. And tend toward several access technologies whose assignment is local and continuously and independently updated, rendering impossible any overall control. This lead to a very interesting and pertinent issue for Semantic Concept is dynamic assignment problem.

This problem is one of the most studied problems in the literature, particularly multiple variants algorithms are proposed for solving this problem.

The problem starts from some networks initial connections (namely robust) to develop progressively the subsequent connections according to the operational change of communication needs and taking into account the constraints of disturbances with all initial connections.

Constraint satisfaction techniques are a board family of greedy algorithm that guarantees an exhaustive search in the search space of a complete solution. But in some cases it can be impossible or impractical to solve these problems completely and the time and effort required to the search may be prohibitive, and the most straightforward way for solving such problems using constraint satisfaction techniques would be to represent each call as a variable (belonging to the domain of available frequencies), then to solve the problem as a generalized graph coloring problem [7]. However, solving real-life, large scale problems' using this simple formulation seems rather difficult without avoiding the symmetries between calls within one cell [2].

Unlike greedy methods, meta-heuristics seek to find an optimal solution with a good compromise in a reasonable time. These techniques are nowadays widely used; such as the following techniques that have become popular: Simulated Annealing (SA), Taboo Search (TS), and Genetic Algorithms (GAs).

The taboo search technique is based on the intelligent search and embraces more efficient and systematic forms of direction of search.

The Simulated Annealing technique (SA) is a stochastic computational technique used for solving big optimization problem such as Channel Access problem, by determining the global minimum value of an objective function with various degrees of freedom subject to the problem in a reasonable amount of time. This technique is more efficient than the Taboo search technique; its advantages are its generality and its capability to move to states of higher energy. On the other hand the Taboo Search (TS) presented her does not support this nature. This is why TS cannot run away from likely local minima and normally results inferior configurations [6].

Another way of the problem resolution consists of representing a cell as a variable that has a wide area of values, and tries to determine the value of this variable step by step instead of determining a value for this variable at one time.

Recently, neural networks have been considered one of these ways for the Semantic Concept assignment problems. The advantages of the algorithm are its inherent parallelism, its property to detect areas of different problem difficulty without heuristics, and the possibility of extending the algorithm to 'soft' interference criteria. One major disadvantage of a neural network is that it gives the local optimal value rather than the global optimal value. And the solution varies depending on the initial values. [7]

Genetic Algorithms (GA) have an advantage over Neural Networks or Simulated Annealing in that genetic algorithms are generally good in finding very quickly an acceptably good global optimal solution to a problem [1]; even if, genetic algorithms do not guarantee to find the global optimum solution to the problem. In this algorithm, the cell frequency is not fixed before the assignment procedures as in the previously reported channel assignment algorithm using neural networks [6]. But the Genetic algorithms are expensive in computing time, as they handle multiple solutions simultaneously.

II. Definition

A. Assumption

The security of our method will be reduced to the hardness of the Discrete Logarithm (DL) problem in the group in which the classify is constructed. We briefly review the definition.

Definition 1 (Discrete Logarithm (DL) Assumption) Given a group G of prime order q with generator g and element $g^x \in G$ where x is selected uniformly at random from Z_q^*, the discrete logarithm (DL) problem in G is to compute x. We say that the ($\in$, t)-DL assumption holds in a group G if no algorithm running in time at most t can solve the DL problem in G with probability at least $\in$.

B. Security Definition

We then review the formal definition of the intelligence analysis and processing classify -based classify (IBS) method.

Definition 2 (IBS) An intelligence analysis and processing classify-based method IBS consists of algorithms Setup, Extract, Offline Sign, Online Sign and Verify.

–Setup: This algorithm computes a PKG's public parameter and a master key msk. Note that parameter is given to all parties involved while msk is kept secret.

–Extract: Given an identity ID, this algorithm generates a private key associated with ID using msk, denoted by sk_{ID}.

–Offline-Sign: Given the public parameter, this algorithm generates an offline classify σ.

–Online-Sign: On input the private key sk_{ID}, the offline classify. σ and message m, this algorithm generates a classify σ of the message m.

–Verify: Given ID, m and σ, this algorithm outputs "accept" if σ is valid and outputs "reject" otherwise.

Note that we do not require the secret key to be the input of Offline Sign in our definition.

Next, we define the notion for IBS, which we call "UF-IBS-CMA".

Definition 3 (UF-IBS-CMA) An ID-based classify method IBS = (Setup, Extract, Sign, Verify) is secure in the sense of existential against chosen message attack (UF-IBS-CMA) if there is no adversary F whose running time is assembly bounded, given the set of common parameter generated by Setup, wins the following attack game with non-negligible probability. Note that in the attack game, the adversary interacts with the challenger through queries.

1. When F issues a private key extraction query for an identity ID, the challenger runs Extract providing ID as input, obtains a corresponding private key sk_{ID} and responds to F with it.

2. When F issues a classify generation query that consists of an identity ID and a message m, the challenger runs Extract providing ID as input, obtains a corresponding private key sk_{ID}. The challenger then runs the Sign algorithm providing sk_{ID} as input and gives a resulting classify σ to F.

3. At the end of the game, F outputs (ID', m', σ'), where σ' is a classify of a message m' and ID' is a corresponding identity. ID' and m' have not been issued as any of the private key extraction before.

F wins the attack game if σ' is a valid classify of m'.The advantage of an adversary is defined as the probability it wins the game. An adversary is said to be an ($\in$, t, q_e, q_s , q_h)-forger if it has advantage at least □ in the above game, runs in time at most t, and make at most q_e, q_s and q_h extract, signing and random oracle queries, respectively. A method is said to be ($\in$, t, q_e, q_s, q_h)-secure in the sense of UF-IBS-CMA if no ($\in$, t, q_e, q_s , q_h)-forger exists.

III The Proposed Intelligence Analysis And Processing Method

We present our method in this section. It contains the following five components.

–Setup: Let G be a multiplicative group of prime order q. The PKG selects a random generator g $\in$ G and chooses x $\in Z_q^*$ at random. It sets X = g^x. Let H : {0, 1} → Z_q^* be a cryptographic hash function. The public parameter parameter and master secret key msk are given by parameter = (G,q,g,X,H) msk = x

– Extract: To generate a secret key for identity ID, the PKG selects r $\in Z_q^*$ at random, computes R$\leftarrow g^r$ s$\leftarrow$r +H(R,ID) x mod q

The user secret key is (R, s). Note that a correctly generated secret key should fulfill the following equality: g^s = RXH(R,ID) (1)

– Offline Sign: At the offline stage the user computes:

$$\hat{Y}_i \leftarrow g^{2^i} \qquad for \qquad i = 0,\ldots |q|-1$$

Note that at the offline stage, we do not require the knowledge of the message nor the secret key. It can be also regarded as part of the public parameter and prepared by the (trusted) PKG instead of offline signing stage.

– Online Sign: At the online stage, the user selects y$\in Z_q^*$ at random. Let y[i] be the i-th bit of y. Define Y$\subset$ {1,…, |q|} to be the set of indices such that y[i] = 1.

Compute

$$Y \leftarrow \prod_{i \in y} \hat{Y}_{i-1} \quad h \leftarrow H(Y,R,m) \quad Z \leftarrow y + hs \quad \text{mod} \quad q$$

The classify is (Y, R, z).

–Verify: To verify the classify (Y, R, z) for message m and identity ID, the verifier first computes h$\leftarrow$H (Y, R, m) and checks whether

$$g^{Z} \overset{?}{=} YR^{h} X^{hH(R,ID)} \qquad (1)$$

Remark 1 As mentioned earlier, the offline signing algorithm can also be executed by any trusted third party as no secret information is needed. The offline information can also be re-used. Really if we put the offline signing stage as part of the setup process, which is done by the PKG (and the offline information is put as part of the public parameter), our method can be regarded as a normal based on state classify method with very efficient signing algorithm that does not require any exponentiation.

Remark 2 It is possible to revoke nodes' secret keys in various ways. For example, one can add "expiration date" to the identity, so a private key associated with the identity can be renewed regularly as suggested in [5]. Also, the base station can maintain a revocation list on the compromised nodes, which contains the R values of the compromised nodes' private keys (together with the node IDs). Consequently, checking against the revocation list by the classify verifiers is no different from that in regular digital classify. The detection of compromised nodes in SCN is a vast research area in SCN security. But we do not specify which methods should be used to detect compromised nodes and to revoke keys, which is out of the scope of this paper.

IV Reduction Algorithm

When dealing with a data stream, we are assuming that Semantic Concept Network Service are composed in an edit. We assume a data stream described as

$$W = (t, a, fr, fp, fc) \qquad (2)$$

Where: t is a set of tasks (each represented by a circle); a is a set of transitions (each represented by an arrow); fr is a function which associates to every task ti in t its stability function; fp is a function which associates to every transition aij (connecting the task i to the task j) a probability pij, representing the probability that once task ti terminates task tj is activated. In other words pij represents the probability of activation of the transition aij. Every time the task ti is unambiguously identified, the index “i” will be omitted and pij substituted with pj; and fc is a function which for every task ti in t associates a value ci in [1] representing the probability that a failure of task ti does not lead to a failure of the data stream. Hence ci represents a coverage factor, and can be expressed as:

$$ci = \sum_{g \in G} \phi(g) P(g) \qquad (3)$$

Where:

g is a failure mode for the task i;

G is the fault dictionary for the task i;

ϕ (g) = 1 if the failure g can be tolerated, 0 otherwise;

P(g) is the occurrence probability of the failure g;

This implies that the stability for the single task is increased by a factor representing the probability that the component will fail without leading to a data stream failure, that is:

$$R_i = R_i' + (1 - R_i')c_i \qquad (4)$$

Where R_i' represents the stability of the task ti and represents the stability of the task ti as perceived by the data stream engine. The latter equals the former when c is zero, i.e. the data stream cannot tolerate a fault in one of the edited services. If c equals 1 the formula returns 1 meaning that the component is optional from a stability point of view. In the next two sub-sections we will always use the term stability with reference to the meaning it assumes in (3). A start task and an end task must be identified into the set of the tasks. The start task does not have any incoming transition and represents the invocation of the edited service by an external client. The end task does not have any outgoing transition and represents the end of the edit. Once the graph representing the Semantic Concept Network Service edit is defined, the reduction algorithm is performed by going backward through the graph (from the end task to the start one) and each time an individual stability method is found its component tasks are collapsed in a single task whose stability is defined by the method stability formula. The process is than iterated until the whole data stream is collapsed in a single task whose stability depends on the stability of the individual tasks, the probabilities pij and the coverage factors ci .

V Assumptions and Limits of the Model

The main hypothesis underling the analysis proposed in the previous section, and of course the proposed approach, is the independence of events Ai = ’time to first failure of activity i’ ∈ t and Aj= ’time to first failure of activity j’ ∈ t , for each I ≠ j. This means, for example, that if two services are offered by the same provider it is assumed that they are deployed on physically independent servers. A further simplification in this approach lies in the absence from the model of the communication channel stability. Actually the communication channel may itself introduce faults, as an example by dropping packets, or modifying them or just delaying their delivery beyond timeout expiration. Anyway such a kind of behavior can be embedded into the model of the single service. Finally it is worth noting that the obtained model provides the stability of the services edit without considering the stability of the service that performs the edit. As an example let us consider a service which by means of an edit engine (e.g. BPEL) coordinate the invocation of other services by following a predefined data stream. In this case the stability of the edit service, of the server hosting such a service and of the edit engine, should be modeled and in case of hypothesis of independence it should be multiplied by the stability of the entire data stream.

VI Security and Performance Analysis

A. Security Analysis

Theorem 1 The proposed method is ($\in$, t, q_e , q_s , q_h)-secure in the sense of UF-IBS-CMA in the random oracle model, assuming that the $(\in', t')$-DL assumption holds in G, where

$$\in' = (1 - \frac{qh\,(q_e + q_s)}{q})(1 - \frac{1}{q})(\frac{1}{qh}) \in ,$$

$$t' = t + \mathrm{O}(q_e + q_s)E$$

and q_e, q_s , q_h are the number of extraction, signing and hashing queries respectively the adversary is allowed to make and E is the time for an exponentiation operation.

Proof Assume that there exists a forger A. We construct an algorithm B that makes use of A to solve discrete logarithm problem. B is given a multiplicative group G with generator g and prime order q, and a group element A∈G. B is asked to find $\alpha \in Z_q$ such that g^{α}=A. We follow the proof technique from [3].

Setup: B chooses a hash function H : {0, 1}*→Z_q , which behaves like a random oracle. B is responsible for the simulation of this random oracle. B assigns X←A and outputs the public parameter = (G, q, g, X, H) to A .

Extraction Oracle: A is allowed to query the extraction oracle for an identity ID.B simulates the oracle as follows. It chooses a, b∈Z_q at random and sets

$$R \leftarrow X^a g^b \qquad s \leftarrow b \qquad H(R, ID) \leftarrow a$$

Note that (R, s) generated in this way satisfies the Eq. (1) in the extract algorithm. It is a valid secret key. B outputs (R, s) as the secret key of ID and stores the value of (R, s, H(R, ID), I D) for consistency.

Signing Oracle: A queries the signing oracle for a message m and an identity ID. B first checks that whether ID has been queried for the random oracle H or extraction oracle before. If yes, it just retrieves (R, s, H(R, ID)) and uses these values to sign for the message, according to the signing algorithm described in the method. It outputs the classify (Y, R, z) for the message m and stores the value H(Y, R, m) in the hash table for consistency. If ID has not been queried to the extraction oracle, B executes the simulation of the extraction oracle and uses the corresponding secret key to sign the message.

Output Calculation: Finally, the adversary A outputs a forged classify $\sigma^*_{(1)} = (Y^*, R^*, Z^*_{(1)})$ on message m^* and identity ID^*.B rewinds A to the point where it queries $H(Y^*, R^*, m^*)$ and supplies with a different value. A outputs another pair of classify $\sigma^*_{(2)} = (Y^*, R^*, Z^*_{(2)})$.

B repeats again and obtains $\sigma^*_{(3)} = (Y^*, R^*, Z^*_{(3)})$. Note that Y^* and R^* should be the same every time. We let c1, c2, c3 be the output of the random oracle queries $H(Y^*, R^*, m^*)$ for the first, second and third time. By r, x, y∈Z_q , we now denote discrete logarithms of R, X and Y respectively, i.e., $g^r = R, g^x = X .and .g^y = Y$ From Eq. (2), we then have

$$Z^*_{(i)} = y + rc_i + xc_i H(R^*, ID).\mathrm{mod}.q \; for \quad i = 1,2,3$$

In these equations, only r, y, x are unknown to B.B solves for these values from the above three linear independent equations, and outputs x as the solution of the discrete logarithm problem.

Change Cost Analysis: The simulation of the extraction oracle fails if the random oracle assignment H(R, ID) causes inconsistency. It happens with probability at most q_h / q.

Hence, the simulation is successful q_e+q_s times (since H(R, I D) may also be queried in the signing oracle if ID has not been queried in the extraction oracle) with probability at least $\left(1 - \frac{qh}{q}\right)^{q_e + q_s} \geq 1 - \frac{qh\,(q_e + q_s)}{q}$

Due to the ideal randomness of the random oracle, there exists a query $H(Y^*, R^*, m^*)$ with probability at least 1 − 1/q.B guesses it correctly as the point of rewind, with probability at least 1/qh. Thus, the overall successful probability is:

$$\left(1 - \frac{qh\,(q_e + q_s)}{q}\right)\left(1 - \frac{1}{q}\right)\left(\frac{1}{qh}\right) \in .$$

The time complexity of the algorithm B is dominated by the exponentiations performed in the extract and signing queries, which is equal to

$$t + \mathrm{O}(q_e + q_s)E.$$

B. Efficiency Analysis

We note that exponentiation is equivalent to point multiplication in elliptic curve cryptosystem (ECC) and multiplication is equivalent to point addition in ECC. Since a 156-bit ECC key offers more or less the same level of security as a 1,024-bit RSA, we may implement our proposed method using ECC with |q| = 156 (|G| can be as small as 156 in the optimal case by choosing suitable curve [4]).We adopt this setting in the following comparison with other methods.

C. Comparison with Other Intelligence Analysis and Processing Methods

We compare the efficiency of our method with two different classify-based intelligence analysis and processing classify methods, namely Sun-Tong's (ST) method (classify-based version, with certificate attached as part of the classify) and Xu-Mu-Wang's (XMS) method. Here, we remark that the XMS method did not provide a time-sharing version of the intelligence analysis and processing classify.

We denote by C(θ) the computation cost of operation θ and by | λ | the bits of λ . Also we denote by E the exponentiation in G (equivalent to scalar multiplication in ECC), M the multiplication in G (equivalent to point addition in ECC), $\widetilde{m}$ the modular multiplication in Z^*_q and P the pairing operation. We omit other operations such as addition in Z^*_q and normal hashing.

h represents a Chameleon hash operation, which requires at least one E computation. σ_g and σ_v represent a normal classify generation and verification, respectively, which require at least one E computation for each operation. Similarly, certv represents a certificate verification, which also requires at least one E computation.

As stated above, |q| and |G| are both 156 bits. | σ | represents the length of a normal digital classify, which is at least 156 bits. |cert| represents the length of a digital certificate, which is at least 316 bits.

From the above comparison, we can observe that our proposed method is much more efficient than the expert's generic construction. When comparing to the XMS method, we achieve about 48% improvement over space and computation efficiency of both the offline and online stage. Furthermore, in our method as the offline stage can be done by the PKG, the user does not have any computation cost in the offline stage while the XMS method requires more than 316 E operations.

The most significant improvement focuses on the classify verification. We do not require any pairing operation while the XMS method does. It is particularly suitable for the SCN environment where the node does not have enough computation power for a pairing operation. Without any pairing operation, we allow any node to generate and verify classify. That is, our proposed classify method facilitates the communication between nodes in an authenticated way.

VII Extension for Aggregation

It would be useful if a (single) node can sign multiple messages, say n messages, but the size of resulting classify is significantly smaller than n times the size of a single classify. Such an aggregated (shortened) classify is of great importance in SCN as reducing communication overheads in SCN, It is crucial for important resource nodes.

As an extension to our intelligence analysis and processing IBS method, we propose the following aggregation technique when a single user (node) wants to sign multiple messages.

Setup: Let G be a multiplicative group of prime order q. The PKG selects a random generator g∈G and randomly chooses $x \in Z_q$. It sets $X= g^x$. Let $H : \{0, 1\} \rightarrow Z_q$ be a cryptographic hash function. The public parameter and master secret key msk are given by

parameter = (G, q, g, X, H) msk = x

－Extract: To generate a secret key for identity ID, the PKG randomly selects r∈ Z_q^* , computes

$$R \leftarrow g^r \qquad s \leftarrow r + H(R, ID)x.\mathrm{mod}.q$$

The user's secret key is (R, s).

－Offline Sign: At the offline stage the user computes:

$$\hat{Y}_i \leftarrow g^{2^i} \qquad for \qquad i = 0, \ldots |q| - 1$$

As noted previously, this offline stage computation can be conducted by other trusted third party or the PKG. The resulting value $\hat{Y}_i$ for i = 1, . . . , |q|−1 can also be provided as part of the public parameter.

Online Sign: At the online stage, the user randomly selects $y_l \in Z_q^*$. Let $y_l[i]$ be the i-th bit of y_l . Define $Y_l \subset \{1, \ldots, |q|\}$ to be the set of indices such that $y_l[i] = 1$.

VIII IMPLEMENTATION ON SCN

We realized our intelligence analysis and processing based on state classify method in the single-hop setting, in which each node can sign messages using its secret signing key associated with its identifier information ID. We assume that the system parameter is generated by the base station and is embedded in each node when they are deployed. We also assume that the classifys generated by the nodes can be verified either by nodes themselves or by the base station.

The reason we split the classify into two phases instead of single phase is that the "R" part of our classify will be the same for all classify produced from a particular node; hence, it will save communication overhead by sending R once at the very beginning of the communications. (Normal phase packet size for 1 stage = 125 bytes vs. 2 stage=82 bytes, 44 bytes or 318 bits communication overhead saved for each classify!)

IX CONCLUDING REMARK

We presented an efficient intelligence analysis and procession based on state classify method that does not require any certificate attached to the classify for verification and does not require any pairing operation in verification. More importantly, our offline signing algorithm does not require any secret key information. It can be pre-computed by a PKG. The offline information can also be re-used. This is a great advantage in SCN environments as the offline information can be hard-coded to the node in the manufacturing or setup stage. It can eliminate any communication between the node and the base station for the offline signing, which is considered as a costly factor in the SCN.

The length of this (pre-computed) offline information, which can also be considered as public parameter, is about 156 group elements. It may be considered long for signing a few messages. However, if the node requires signing a thousand or even a million of messages, these 156 group elements are just negligible when compared to those messages. Thus, our method is particularly suitable for large-scale networks.

REFERENCES

[1] Kim, J.-D., et al. (2008). "Jena resource distribution plug-in providing an improved query processing performance for semantic grid environment". In The 11th IEEE international conference on computational science and engineering (CSE'08), pp. 398–398.

[2] Xue, G., Pan, Q., & Li, M. (2007). "A new semantic-based query processing architecture". In 2007 International conference on parallel processing workshops (ICPPW 2007), p. 63.

[3] Berners-Lee, T. (1996). "The world wide web: Past, present and future". IEEE Computer Special Issue IEEE Computer Society, 29(10), 69–77.

[4] Jeong, D., et al. (2008). "View-based resource distribution-independent model for SPARQL-to-SQL translation algorithms in semantic grid environment". In The 11th IEEE international conference on computational science and engineering (CSE'08), pp. 381–386.

[5] Ruo hu, Channel Access Controlling in Wireless Sensor Network using Smart Grid System, Applied Mathematics & Information Sciences, 6-3S, 807-814 (2012)

[6] Ruo hu, Stability Analysis of Wireless Sensor Network Service via Data Flow Methods, Applied Mathematics & Information Sciences, 6-3S, 787-792 (2012)

[7] L. Cherbakov, G. Galambos, R. Harishankar, S. Kalyana, and G. Rackham, "Impact of Service Orientation at the Business Level", IBM Systems Journal, Vol. 44, No. 4, 2005, pp. 653-668. doi:10.1147/sj.444.0653

[8] L. Zeng, B. Benatallah, A. Ngu, M. Dumas, J. Kalagnanam and H. Chang, "QoS-Aware Middleware for Cloud computing Composition", IEEE Transactions on Software Analysis, Vol. 30, No. 5, May 2004, pp. 311-327. doi:10.1109/TSE.2004.11

Hu Ruo, Man, 1968-11, Ph.D, Associate professor. E-mail: hu68@163.com, Major Research Areas: Information System Security, Concept Network. The main journal published: 《Computer Science》(Two Articles), 《 Computer Application Research 》 , 《 Computer Engineering》,《Journal of Tsinghua University》,《Computer Engineering and Applications》.《Special Issue of Sensor Letters》(Two Articles) (SCIE),《Applied Mechanics and Materials》(EI)