International Journal of Grid and Distributed Computing
Vol. 9, No. 1 (2016), pp.135-144
http://dx.doi.org/10.14257/ijgdc.2016.9.1.14

A Novel Crawler Based on Loginning Simulation for Weibo Social Network

Ling Xing[1, 2], Ling Jiang[3] and Bao Peng[4]

[1]*School of Information Engineering, Southwest University of Science and Technology, Mianyang, 621010, China*
[2]*Robot Technology Used for Special Environment Key Laboratory of Sichuan Province, Mianyang, 621010, China*
[3] *Department of Mathematics and Computer Science, Wuyi University, Wuyishan, 354300,China*
[4] *School of Electronic and Communication, Shenzhen Institute of Information Technology, Shenzhen, 518172, China*
xingling_my@163.com

Abstract

With the rapid development of Weibo, which is the most popular microblog in china, more and more attention was paid to relative studies about it. With the objective of gathering precise information data from Weibo, which is the groundwork of these researches, a novel high efficient Weibo crawler (WCrawler) based on loginning simulation is designed. The priority evaluation is described to ensure the correlation between entires. MD5 is introduced to check for duplicates of URL crawled. Experiments demonstrate that the novel crawler has an efficiency and integrity of information collecting compared with API crawler. In addition, we present a summary of the data that collected from Weibo social network by WCrawler.

Keywords*: Weibo Crawler, Loginning Simulation, Web Information Extraction*

1. Introduction

With the tremendous growth of the Web application and mobile communication, Weibo has attracted more and more attention of users, researchers and enterprises. It is the first step of researchers to collect Weibo data. It has been analyzed for several years to extract useful information, and the open microblog API [1] interfaces has been used widely to fetch data in most studies, which has the advantages . Unfortunately, the method over API has limitations of quantity and frequency, and many researches do need a large amount of different data. Therefor it is not suit for many applications, such as user recommendation.

Many crawling frameworks designed for general websites are introduced to fetch data from Weibo pages. Compared to microblog API, this technique has the following advantage: (1) Higher coverage percentages of microblog data. (2) Fewer limitation on Weibo platform. (3) More operable. To overcome those new challenges brought by Weibo, protocol-driven and event-driven are used [2]. A script-based function and parameters of the hot spot detection mechanism [3] are proposed, and the novel mechanism has advantage of reducing the duplication of information crawling. In 2012, Dayong Shen [4] designed a incremental crawler based on the classic multi-producers and multi-consumers model, which can collect realtime microblog information efficiently.

However, since the widespread use of Ajax [3] technology and security authentication, those existing microblog crawlers also face serious challenges. On one hand, general crawler may miss some important information owning to scroll Weibo pages. On the

ISSN: 2005-4262 IJGDC

other hand, with the revolution of Weibo's login mechanisms, crawler could visit nothing without logining in Weibo. So solving the problem of simulating login has become the priority issue. In this paper a novel WCrawler is designed, which could simulate Weibo login and get authorization based on Base64 and RSA2 [5-7]. It also achieves the successful information extraction and storage.

2. Data Specification and Priority Evaluation

In order to effectively implement information extraction for Weibo social network, the data specification and the priority evaluation of users crawled for Weibo user recommendation are described as follows.

2.1. Data Specification

The necessary attributes for user recommendation are listed in Table 1. *User_Info* is constructed to describe the basic information of user. R1 stands for *Relation_Info*, which is consisted of UIDs of root user and child user, and the relationship flag. *Status_Info* includes the basic interaction metadatas of Weibo, especially, S3 and S7 are effectual to reflect the influence between users.

Table 1. User Metadata Specification

Name	Content And Explanation
User_Info	U_1 (UID); U_2 (screenName); U_3 (verified); U_4 (statusCount); U_5 (fansCount); U_6 (friendsCount); U_7 (sex); U_8 (path);U_9 (mutual_fans);
Relation_Info	R_1 (relationship)
Status_Info	S_1 (MID); S_2 (creatTime; S_3 (isForward); S_4 (commentNum); S_5 (forwardNum); S_6 (praiseNum); S_7 (isMention)

2.2. Priority Evaluation

According to triadic closure, the second-level and third-level contacts will be gathered by WCrawler and recommended to the target user. $F_n(u)$ stands for the n-level contacts, thus $F_1(u)$ is the set of followers of u, and the higher-level contacts are defined as follows. The relevant users $R\_U(u)$ is described as equation (3),

$$F_2(u) = F_1(F_1(u)) - F_1(u) - \{u\} \tag{1}$$

$$F_3(u) = F_2(F_2(u)) - F_2(u) - F_1(u) - \{u\} = F_2(F_2(u)) - F_1(F_1(u)) \tag{2}$$

$$R\_U(u) = F_2(u) \cup F_3(u) \tag{3}$$

Owning to the number of users introduced from the same group is larger, it is necessary to evaluate the priority of them to ensure WCrawler has always focused on the correlative users. The influences came from U_4, U_5 and the dossemination of *Status_Info* are vital to priority evaluation as shown in (4), which are represented as $F(u)$, $S(u)$ and $D(u)$ respectively and defined as a set of the following equations.

$$P(u) = \omega_1 \bullet F(u) + \omega_2 \bullet S(u) + \omega_3 \bullet D(u) \tag{4}$$

$$F(u) = \lg(U_5(u)) / \lg(\frac{1}{n}\sum_{i=1}^{n} U_5(i)) \tag{5}$$

$$S(u) = U_4(u) / (\frac{1}{n} \sum_{i=1}^{n} U_4(i)) \tag{6}$$

$$D(u) = \mu_f \bullet \frac{\frac{1}{m} \sum_{p=1}^{m} S_5(up)}{\sum_{i=1}^{n} \sum_{j=1}^{m} S_5(ij) / n \bullet m} + \mu_c \bullet \frac{\frac{1}{m} \sum_{q=1}^{m} S_4(uq)}{\sum_{i=1}^{n} \sum_{j=1}^{m} S_4(ij) / n \bullet m} \tag{7}$$

3. Design and Implementation of WCrawler

For purpose of collecting information for user recommendation, WCrawler based on logining simulation is proposed and realized, and the architecture for the novel crawler is displayed in Figure 1. How WCrawler works is described as follows.

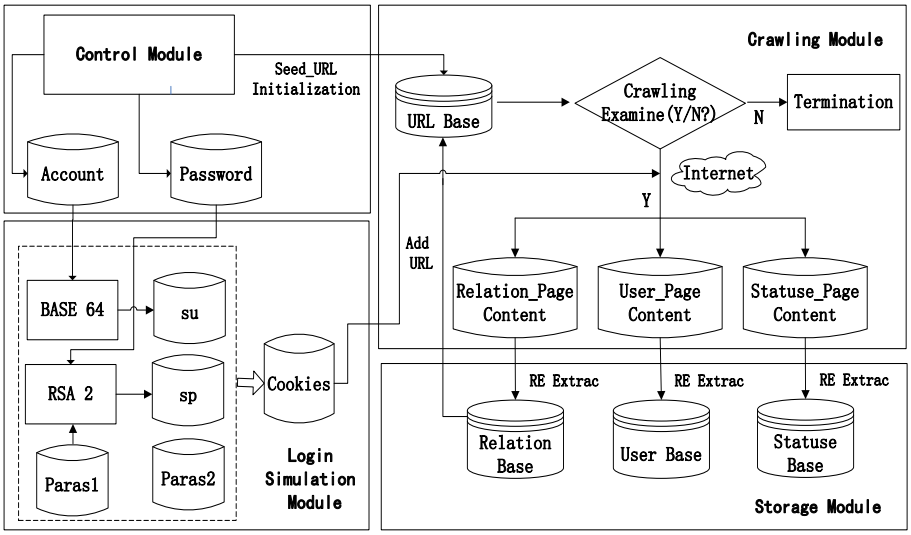

Figure 1. The Architecture of WCrawler

3.1. Control Module and Storage Module

Control module has the responsibility to initialize seeds for WCrawler, provide parameters for login simulation module and terminate the crawler in some special situation. The storage module adopts MySQL as the database to store data information.

3.2. Login Simulation Module

It is essential and obligatory for WCrawler to realize the login simulation model. Many parameters are needed in the process to construct the login information packets, and some important parameters are chosen to list in Table 2.

The process of login simulation is the inverse of the data packet capture, including creating a request object, loading the requested data, login simulation and sending the request message, getting the responses and extracting relative parameters, and constructing the cookies. HttpClient adds extra functionality by providing a feature-rich and fully functional Java package for accessing resources through HTTP. It also has built-in functions to store cookies and handle certificates. These advantages make HttpClient be adopted for login simulation module. The process of login simulation is described in detail as follows.

Table 2. Parameters for Login Simulation Module

Name		Value
Paras1	pubkey; severtime; nonce	Extracting from HTTP Response
Paras2	pcid; rsakv	
	service	miniblog
	encoding	UTF-8
	pwencode	rsa2
	ssosimplelogin	1
	useticket	1
	entry	weibo
	URLs	http://www.weibo.com, ect.
su		Base 64(account)
sp		RSA 2(password+Paras1)

1) Some parameters could be returned from Weibo server, such as "servertime", "nonce", "pubkey" and "rsakv", after getting a demo account name and corresponding password form control module, and sending the logining information packets to Weibo server, where the information packets are Base64 security encryption.

2) RSA encryption. According to step one, we have four parameters. The first parameter "servertime" stands for the last time server provided service. The second parameter "nonce" is a random string generated by server. What's more, "pubkey" and "rsakv" could be set to fixed value, which are important to RSA encryption. It is necessary to combine Para1 and password which is provided in control module, and then encrypte them with RSA2 to get a unique and irreversibility secret key, which will be used in integrality and identity authentication.

3) Getting Cookies. Parameters in Table 2, most of them are acquired by combining the get request and regular expression (RE). After constructing a form with those parameters, WCrawler sends a request to server through the form post. And the server ticket and authentication URL could be acquired from the HTTP response. The correct returned Cookies is available by connecting to the authentication URL through get request.after the heading number, not a colon.

3.3. Crawling Module

As the core unit of WCrawler, the crawling strategy is described in Figure 2 detailedly. It starts running with a seed URL to download the corresponding user's home page source from the Weibo server. And then fetches and extracts *User_Info*, *Relation_Info*, Repost_Info and *Status_Info* by visiting different corresponding URLs and using regular expression, where the URLs are constructed with UID.

In order to check for duplicates, we have keep a MD5 [8,9] hash of URL, which will process a variable-length URL into a fixed-length output of 128 bits. The process is described as follows. 1) Padding the URL message with a single 1-bit and several 0-bit, until the total length in bits is congruent to 448 modulo 512 ($N \bullet 512 + 448$). 2) Filling up the remaining 64 bits with the length of the original URL ($(N+1) \bullet 512$). The result could divided into $(N+1) \bullet 16$ blocks with 32-bit. 3) Initialize MD5 parameters with four 32-bit certain diced integers, denoted a, b, c, and d. The processing of a block includes four rounds with nonlinear functions defined as (8). Different Mi and Ki are used in every round shown as (9), which are calculated with a 32-bit constant, where $<< s$ means a left bit rotation by s places. Cascading above steps, a 128-bit hash value for URL is generated.

$$
\begin{aligned}
&Round\_1: F(b,c,d) = (c \wedge b) \vee ((\neg b) \wedge d) \\
&Round\_2: G(b,c,d) = (b \wedge d) \vee (c \wedge (\neg d)) \\
&Round\_3: H(b,c,d) = b \oplus c \oplus d \\
&Round\_4: I(b,c,d) = c \oplus (b \vee (\neg d))
\end{aligned} \tag{8}
$$

$$
\begin{aligned}
&FF \rightarrow a = b + ((a + F(b,c,d) + M_i + K_i) << s) \\
&GG \rightarrow a = b + ((a + G(b,c,d) + M_i + K_i) << s) \\
&HH \rightarrow a = b + ((a + H(b,c,d) + M_i + K_i) << s) \\
&II \rightarrow a = b + ((a + I(b,c,d) + M_i + K_i) << s)
\end{aligned} \tag{9}
$$

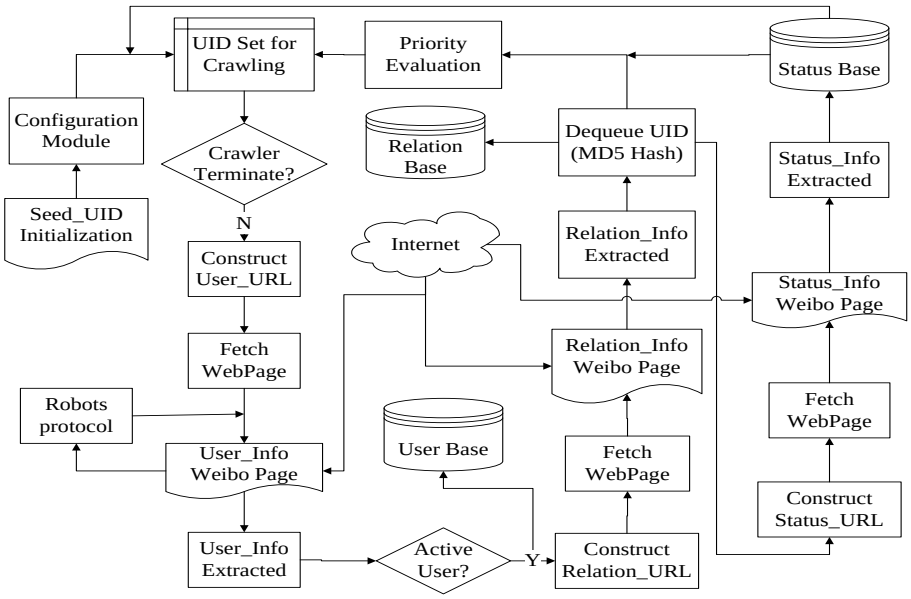

Figure 2. Crawling Strategy of WCrawler

Owning to scroll pages, it is difficult to fetch integrated information of Weibo status. In this situation, the WCrawler downloads and extracts 15 statuses as a group in one status page, and then combines several groups data to analyses status information.

4. Experimental Results and Analysis

In this section, we analysis the performance of WCrawler, and present some characteristics of the data that we obtained from Sina Weibo social network using WCrawler described in the previous section. not a colon.

4.1. Performance of WCrawler

In this paper, U1=2569036344 is token to construct the seed URL to start the WCrawler system, which is written in Java language that is compiled in the environment of JDK 1.7.0. In order to evaluate the performance of WCrawler, the crawler with Weibo API is chosen to compare with presented crawler system. What's shown in Figure 3 are the simulation results of crawling speed with WCrawler and API crawler. It is observed that the speed of WCrawler is faster than API crawler to extract *User_Info*, where it approximately takes 20s for the former to crawler 100 user's *User_Info* and 55s for the latter. Because of the limitations of quantity and frequency, the speed of API crawler has a larger fluctuation than WCrawler, and it is unable for API crawler to collect integrated

Relation_Info and *Status_Info*. What's more, it takes more time apparently to collect *User_Info*,*Relation_Info* and *Status_Info* in the first group, which is due to the so called warming up stage. Because the crawled numbers of relations and statuses are depended on the different user, different group simulation results are unstable.

Figure 3. Performance Comparison between WCrawler and API

4.2. Analyzing the Crawled Data

The same to other social networks, users could also connect to other interested users by the functionality of *follow* on Weibo. So user network has the characteristics of a directed graph. For a user, the set of people who follow him is called *Fans*, and the set of *Friends* is consist of the people he follows. Figure 4 presents an alternative view of U_4 , U_5, U_6 and U_9 distribution on log-log scale. As shown in figure, the distributions follow the discrete power law : $p(\tau) = C\tau^{-\alpha}$, where the power exponents are listed as follows: $\alpha_{U_4} = 0.4185$, $\alpha_{U_5} = 0.7621$, $\alpha_{U_6} = 1.1806$, $\alpha_{U_9} = 1.0624$. What's more, the power-law distributions of U_5, U_6 have the nutation nature, and the power-law distributions of (b) and (d) dispaly the heavy-tailed nature.

In addition, the empirical complementary CDFs of U_4 , U_5, U_6 and U_9 on a log-log scale are shown in Figure 5. The x-axis stands for the value of U_4 , U_5, U_6 or U_9, and y-axis equal to the CDFs, where $F_X(x) = p(X \le x)$. The area with greater slope of CDF curve stands for the concentration range of x's value. It also indicates that only a few users have a large fans, friends or statuses, and the maximum of U_5 is much higher than U_4, U_6 and U_9.

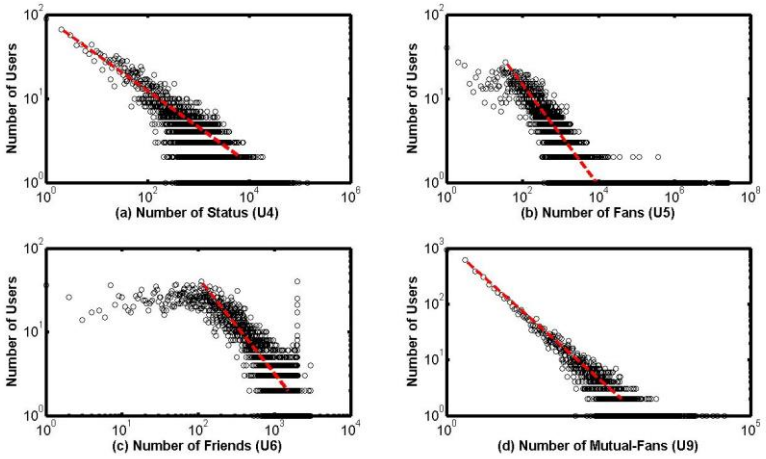

Figure 4. The Power-Law Distribution of U4 , U5, U6 and U9

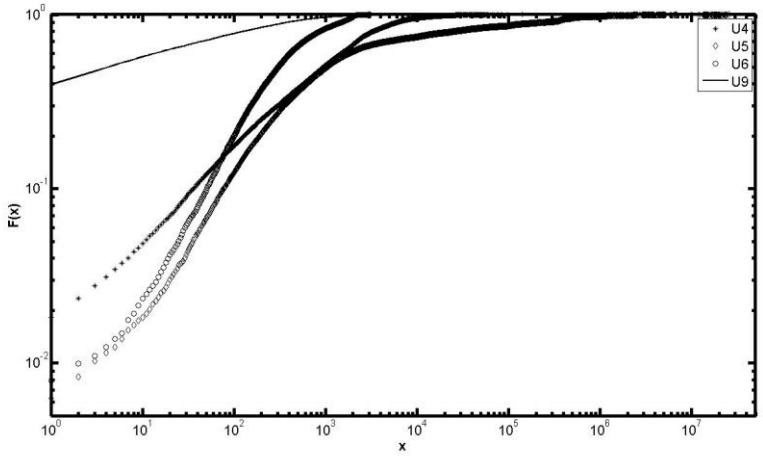

Figure 5. The CDF Curves of U4 , U5 U6 and U9

In Figure 6 we plot U_5 versus U_4 and U_6, which are collected by WCrawler. As shown in the Figure (b), the first set of users represent the majority users in real-life, who have a small number of U_5 and U_6. The second set of users could well be consist of verified users and celebrities, who display a small number of friends and a larger number of fans. They have a lot of followers not only in reality life, but also in Weibo social network. On the contrary, the third set of users have follow a lot of people, but only a small number of users follow them. According to the relationship between U_4 and U_5, users could divide into four sets as shown in Figure 6 (a). The first set of users who display a lager number of U_4 and a higher number of U_5 than majority users, we conjecture that the statuses of such user are probably have influence and ability to attract fans. The other three sets of user have the approximate distribution with Figure 6 (b).

Moreover, we show the distribution of the published intervals of Weibo on a log-log scale. As shown in Figure 7, the distributions also follow the discrete power law with the power exponents $\alpha = 1.6434$. These results show that the collected data accords with the fact situation.

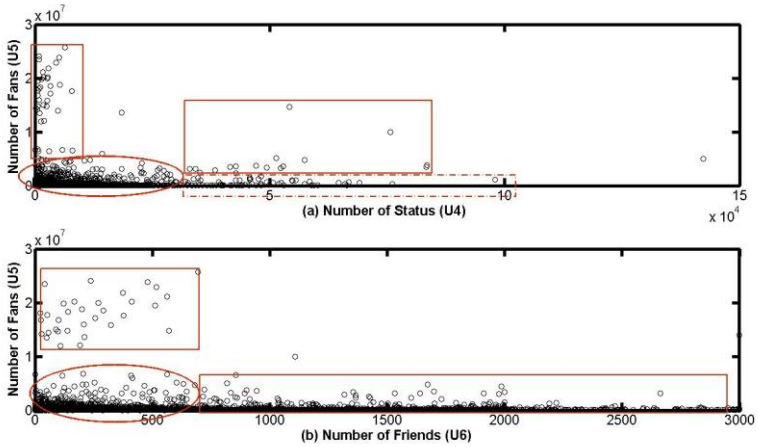

Figure 6. The Distribution of U4 - U5 and U6 - U5

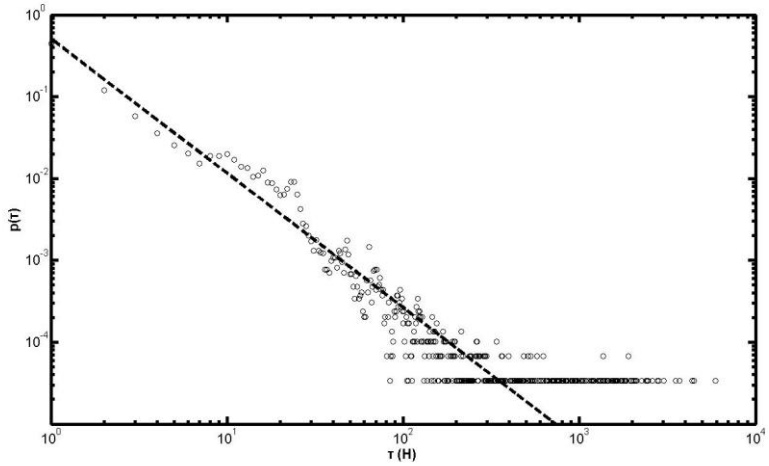

Figure 7. The Distribution of the Published Interval of Statuses

5. Conclusions

In this paper, we introduce loginning simulation to collect information based on the Weibo social network, design and implement the WCrawler. MD5 and priority evaluation are used to improve the efficiency of WClawer and the correlation between users. The strategy of extraction by groups is adopt to overcome the challenge brought by scroll pages. Experimental results demonstrate that the novel crawler can collect realtime Weibo information precisely, and WCrawler offers advantages over API crawler both in efficiency and integrity. On one hand, WCrawler could save sbout 35s than API crawler to crawler 100 user's *User_Info*. On the other hand, the performance of WCrawler is more stable than API crawler. We also analysis and present some characteristics of the data collected. And the distribution of fansConut, friendCount, statusCount, mutual_fans and the published interval of statuses are closely approximate power-law distributions with heavy-tailed and nutation nutures. Those results presented are vitial to analysis Weibo social network, and show that the collected data accords with the fact situation.

Acknowledgments

This research is partially supported by National Nature Science Foundation of China (Grant No. 61171109), Applied Basic Reasearch Programs of Sichuan Science and Technology Department (Grant No. 2014JY0215, No. 13ZA0161,No.13zd3107), and Basic Research Plan in SWUST and Shenzhen City (Grant No.12zxwk01, No.13zx9101，No. JCYJ20130401100512995). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] J. Lian, X. Zhou, W. Cao and Y. Liu, "SINA microblog data retrieval", Journal of Tsinghua University Science and Technology, vol.51, no.10, **(2011)**, pp.1300-1305.
[2] X. H. Yuan, S. S. Zhou, "Research and implementation of the technology supporting microblog data collection based on web crawler", Proceedings of International Conference on Automatic Control and Artificial Intelligence, **(2012)**, pp.1674-1677.
[3] D. Cristian, F. Gianni, K. Donald, M. Reto and Z. Chong, "AJAX crawl: Marking AJAX applications searchable", Proceeding of 25th IEEE International Conference on Data Engineering, **(2009)**, pp.78-89.
[4] D. Shen, H. Wang, J. Cao, P. Li and Z. Jiang, "A high efficient incremental microblogg crawler: design and implementation", Journal of Information & Computational Science, vol.10, no.6, **(2013)**, pp.1731-1747.
[5] S. Josefsson, "The base16, base32, and base64 data encodings", The Internet Society, **(2006)**.

[6] M. A. Ahmad, I. F. Al and H. M. Ahmad, " Protection of the Texts Using Base64 and MD5", Journal of Advanced Computer Science and Technology Research, vol.2, **(2012)**,pp.22-34.
[7] J. S. Patel, V. M. Chavda. "Security Vulnerability and Robust Security Requirements using Key Management in Sensor Network", International Journal of Grid & Distributed Computing, vol.7, no.3, **(2014)**, pp.23-28.
[8] K. Ah, C. Mary, Z. Wang, and D. Shubra, "Security analysis of MD5 algorithm in password storage", Instruments, Measurement, Electronics and Information Engineering, vol.347-350, **(2013)**, pp.2706-2711.
[9] C. Ng, T. Ng and K. Yip, "A unified architecture of MD5 and RIPEMD-160 hash algorithms", Proceeding of the 2004 International Symposium on Circuits and System, **(2004)**, pp.889-892.

Authors

Ling Xing, (1978-), She received the Ph.D. Degree in Communication and Information System from Beijing Institute of Technology, Beijing, China, in 2008. She is a Professor with communication engineering, School of Information Engineering, Southwest University of Science and Technology, China. Her research interests include Information Semantic Processing and Information Networks.

Ling Jiang, (1990-), She received the Master's Degree in Southwest University of Science and Technology. She is a assistant with communication engineering, Department of Mathematics and Computer Science, Wuyi University. Her major intersts include Data Mining and Computer Social Network.

Bao Peng (1979-), He received the PhD degree in Information and communication engineering from the Institute of Electronic Information Engineering, Harbin Institute of Technology (HIT) in Harbin, Heilongjiang, 2009. His research interests are in the areas of Sensor Networks, Distributed Systems, and Information Management.

