

User Authentication Scheme In Cloud Computing

Hemant Agrawal, Ajay Thakur, Rajan Slathia, N. Jeyanthi

School of Information Technology and Engineering

VIT University, Vellore,

Tamilnadu, 632014

hemant.agrawal2014@vit.ac.in, ajay.thakur2014@vit.ac.in, rajan.slathia2014@vit.ac.in

Abstract

Cloud computing is a way of computing which enables the user (individual user) or group (companies or organization) to on-demand access the available universal computing resources on sharing basis which means that it do not have to own the resources (server, storage and application) and can use multiple servers through internet rapidly and conveniently. Cloud is also suffering from many issues which needs to be solved and one of them is user authentication, as cloud is vulnerable to unauthorized usage with the exponential increase in the cloud users. Some user authentication schemes are Two factor authentication (2FA), Elliptic Curve Cryptography (ECC), Ticket authentication, MD5, MD6 (message-digest) Graphical passwords and many more. This paper aims at development of a new user authentication scheme for identification of legitimate users to secure the cloud resources. This paper tries to existing schemes and overcome some of the loop holes present in them regarding the user authentication in cloud environment.

Keywords: Cloud Computing, User Authentication, Access Control, Secure Resources, Security Challenges, Kerberos, MD6.

Introduction

Cloud computing is a type of computing that depends on distribution of computing resources [2] other than having local servers or personal devices to handle software. These administrations are extensively separated into three classes:

- Infrastructure as a Service
- Platform as a Service
- Software as a Service

Infrastructure as-a-service like Amazon Web Administrations gives the client virtual server cases and capacity, and also application program interfaces (APIs) [8]

that permit the client to begin, stop, get to and design their virtual servers and capacity.

Platform-as-a-service in the cloud is an arrangement of software development tools encouraged on the supplier's side. Developers build applications on the producer's stage over the Web. Planners need to comprehend that beginning now, there are not measures for interoperability or information minimization in the cloud.

In the Software-as-a-service, the supplier supplies the hardware, the software and meet with the customer through a web pages. Associations can be anything from online email to stock control and database get prepared. Since the association supplier has both the application and the information, the end client is allowed to utilize the association from wherever.

A cloud association [9] has three unique attributes that distinctive it from standard empowering. It is sold on venture, mostly by the moment or the hour; it is versatile and the association is completely directed by the supplier (the purchaser needs simply a PC and Web access). Basic movements in virtualization and passed on enrolling, and besides enhanced access to brisk Web and a feeble economy, have restored energy to pass on transforming.

A cloud [10] can be personal or open. A public cloud gives associations to anybody on the Web. (At this minute, Amazon Web Administrations is the best open cloud supplier.). Personal or open, the aim of cloud computing is to give fundamental, versatile access in taking care of assets and IT benefits.

Solid client validation is the vital necessity for cloud computing that limit unlawful access of cloud server. In this respect, this paper proposes a solid client confirmation structure by setting aside a onetime password (OTP), message Digest-6(md-6) and Kerberos; where client authenticity is emphatically confirmed before enter into the cloud. The proposed structure gives personality administration, common confirmation, session key foundation between the clients and the cloud server. A client can change his/her secret word, at whatever point requested. Moreover, security examination understands the achievability of the proposed skeleton for distributed computing and attains effectiveness.

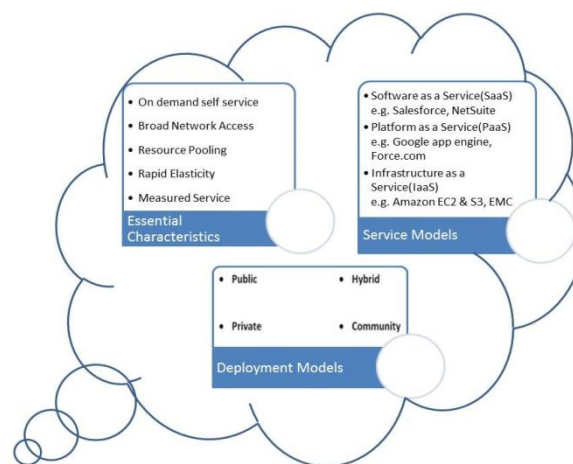


Figure 1: General Cloud Computing Model

Related Work

Cloud server building design is utilized as a part of distributed computing in huge scale. Client verification in cloud computing assumes an exceptionally indispensable part in approving the specific client, whether the client is genuine or not.

The majority of the mainstream remote confirmation methods were recommended by Lamport in 1981. In these techniques, server stores both User_id and watchword in hash table for confirmation. Numerous hash capacities were utilized, which naturally creates some arbitrary secret key. A portion of the current secret key validation plans were utilized as a part of Smartcard to keep from the assault. Keeping in mind the end goal to make a protected use of administrations gave by the cloud, Client validation model can utilize distinctive watchword procedures like

- 1) Straightforward content secret word
- 2) Graphical secret word verification
- 3) 3d secret word object.

The shortcoming of secret word confirmation framework is, it can be broken and is defenseless against assault. Graphical secret word obliges less or rise to memory space than text based watchword. Though graphical watchword oblige substantial space and time. 3d watchword having it limits. Some proposed validation plans focused around sending SMS doesn't promise the conveyance of SMS on time. So in this manner, this paper proposes another client verification plan where the secret word created with the assistance of different calculations will be sent to the client's enlisted portable number and Email_id. The client will be furnished with the alternative of changing the versatile number in the event that he/she loses his/her portable.

Proposed System

User authentication scheme in cloud computing is necessary to keep the integrity of the cloud. In this paper user authentication is checked at login level which is very important.

The paper introduced a modified scheme in which all the users keep their User ID for login which is assigned at the time of signup. During sign up users have to verify their identity by providing there any of identity card (Passport, License, Voter ID, Aadhar Card, Pan card) and also some security questions answer to make identity unique and secure. User also has to provide his existing mobile number and e-mail so that he can receive his password.

After successful sign up every time user try to login he has to enter his ID after that he has 2 options to receive his OTP either on his cell or his mail ID as he received his password he enters that OTP and move to next step of login process.

User faces 1 of the security question chosen randomly among the questions he filled at the time of sign-up, user has to answer the question after verification user is able to login to his personal account to access cloud. For password encryption it is using MD6 encryption technique which encrypts password in up to 512 bits to transmit over medium.

For session providing it is using Kerberos which enables user to login through only 1 system at a time Kerberos issues a unique ticket to a particular user so that another

user or same user cannot login through multiple systems at a time. Kerberos also manages sessions so if user is idle for a long time then it will automatically expire his session and user has to login again. In case if user lost his mobile phone and also won't have access to his mail id in that case user can simply call the help center to receive his OTP after verification.

The above scheme makes use of following authentication algorithms:

- One time password(OTP)
- MD-6 Cryptographic hash function
- Kerberos

A one-time password (OTP) is a password [12-15] that is legitimate for stand out login session or transaction. OTPs maintain a strategic distance from various weaknesses that are connected with customary (static) passwords. The most imperative deficiency that is tended to by OTPs is that, as opposed to static passwords, they are not helpless against replay assaults. This implies that a potential interloper who figures out how to record an OTP that was at that point used to log into an administration or to direct a transaction won't have the capacity to misapply it, since it will be no more legitimate.

On the drawback, OTPs are troublesome for people to retain. In this manner they require extra engineering to work OTP era calculations normally make utilization of pseudo randomness or haphazardness. This is vital in light of the fact that else it would be not difficult to anticipate future OTPs by watching past ones. Cement OTP calculations change significantly in their subtle elements. Different methodologies for the era of OTPs are recorded beneath:

- Taking into account time-synchronization between the verification server and the customer giving the secret key (OTPs are substantial just for a brief time of time)
- Utilizing a scientific calculation to produce another watchword focused around the past secret key (OTPs are viably a chain and must be utilized as a part of a predefined request).
- Utilizing a scientific calculation where the new secret word is focused around a test (e.g., an arbitrary number picked by the confirmation server or transaction subtle elements) and/or a counter.

The Md6 Message-Digest[1][4][5][7] Calculation is a cryptographic hash capacity. It utilizes a Merkle tree-like structure to consider monstrous parallel calculation of hashes for long inputs. Creators guarantee an execution of 28 cycles for every byte for Md6-256 on an Intel Center 2 Team and provable safety against differential cryptanalysis. Speeds in over abundance of 1 GB/s have been accounted for to be feasible for long messages on 16-center CPU building design.

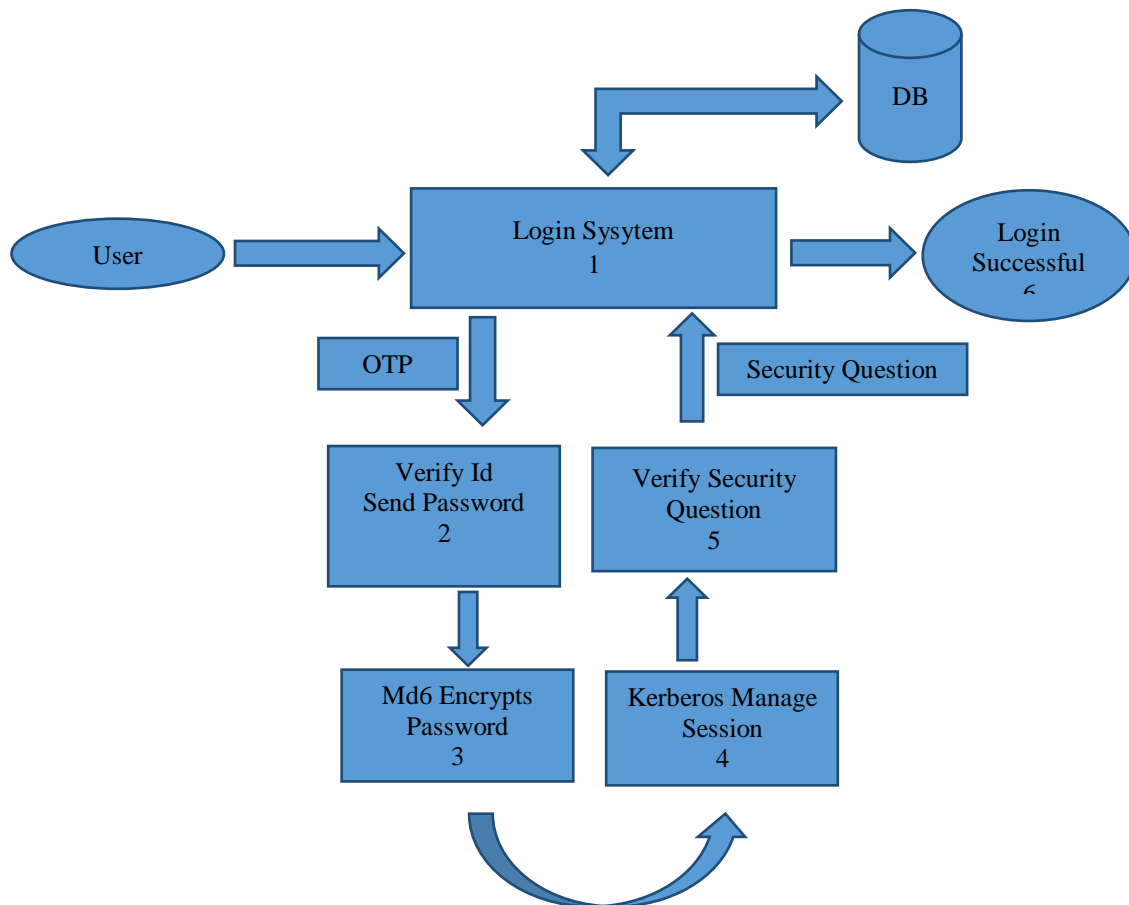


Figure 2: Proposed Model Architecture

Kerberos:

The Kerberos Verification Framework [11] makes use of an arrangement of encoded messages to demonstrate to a verifier that a customer is running in the interest of a specific client.

1) Kerberos Encryption

Kerberos verification helps in finding whether a customer is running in the interest of a specific client, a more exact explanation is that the customer has knowledge of secret key that is known by just the client and the confirmation server. In Kerberos, the client's encryption key is gotten from and ought to be considered a password.

2) The Kerberos Ticket

The Kerberos ticket is issued by a validation server, encoded utilizing the server key. The ticket is not sent specifically to the verifier, however is rather sent to the customer

who advances it to the verifier as a major aspect of the application demand. Since the ticket is scrambled in the server key, known just by the confirmation server and expected verifier.

3) Application request and response

On receiving the request, the verifier converts the ticket, deletes the session key, and puts session key into consideration to unscramble the authenticator. Also the verifier additionally checks the timestamp to check that the authenticator is new.

4) Authentication appeal and reaction

When a client wishes to make an association with a specific verifier, the client uses the authentication request and response to obtain a ticket and session key through the authentication server.

The authentication server responds with the session key, the assigned expiration time, the random number from the request, the name of the verifier, and other information from the ticket, all encrypted with the user's password registered with the authentication server, together with a ticket containing similar information.

5) Generating more tickets

This protocol allows a client with information of the user's password to attain a ticket and session key to prove its identity to any validator registered to the authentication server. The client's password must be displayed each one time the client performs confirmation with another verifier.

6) Protecting application data

The result of the Kerberos is the exchange of the session key between the customer and the server.

Table 1: Study of Various Authentication Schemes

Parameters	Graphical Passwords	2FA:Hash PWD +ZKP	Ticket authentication	MD5 Hash	MD6 Hash
Identity management	Yes	Yes	Yes	Yes	Yes
User Privacy	Yes	Yes	Yes	Yes	Yes
Mutual Authentication	Yes	Yes	Yes	Yes	Yes
Replay attack	Yes	Yes	Yes	Yes	Yes
Man in the middle attack	Yes	Yes	No	Yes	Yes
Denial of service	Yes	No	Yes	Yes	Yes
Masquerade attack	Yes	Yes	Yes	Yes	Yes
Password guessing attack	No	Yes	Yes	No	Yes
Insider attack	No	Yes	Yes	No	Yes
Anonymity	No	Yes	No	Yes	Yes
Computational cost	Low	Low	Low	High	High
Shoulder surfing attack	No	Yes	Yes	Yes	Yes
Phishing attack	No	Yes	Yes	No	Yes
Password change phase	Yes	No	Yes	No	Yes

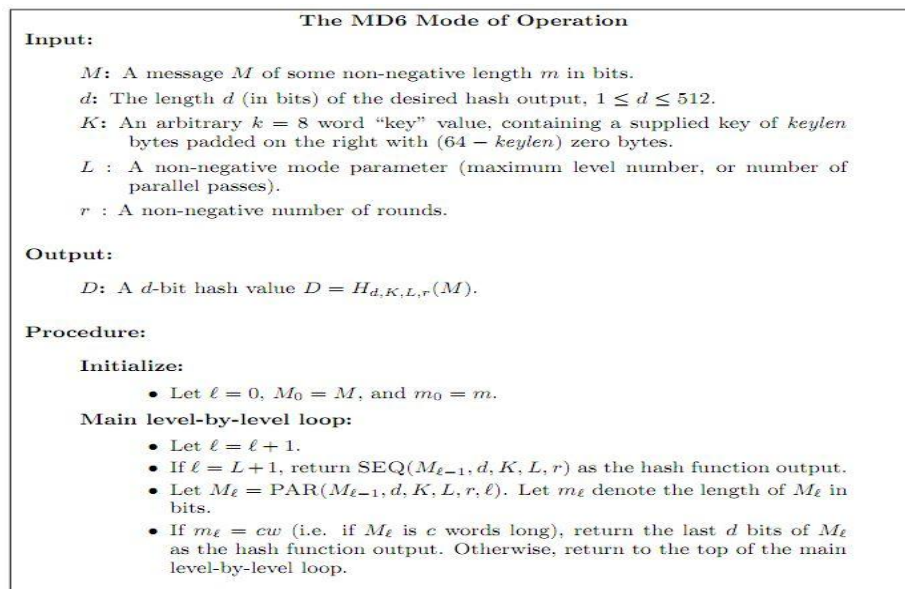


Figure 3: The MD6 Mode of Operation

(Courtesy: [1] *The MD6 hash function-A proposal to NIST for SHA-3, October 27, 2008*)

Cryptanalysis

MD6: Differential cryptanalysis

The workload for a standard differential assault against MD6 is provably bigger than the workload for a basic \birthday assault,"for all NIST indicated yield sizes. Truth be told, stronger result the workload for a standard differential assault against MD6 is provably bigger than the workload for a basic \birthday assault," for all yield sizes.

A standard differential assault won't be viable against MD6 there are no differentialways with sufficiently high likelihood to make such an assault more productive than a basic birthday headed assault searching for crashes. It thinks about this as an extremely huge result, subsequent to in vast part it has been the accomplishment of differential assaults against hash works that roused NIST to compose the SHA-3 hash capacity rivalry. The way that MD6 is not helpless against standard differential assaults is extremely engaging. Obviously, there may be nonstandard differential assaults (e.g. that utilized different types of summed up differentials, as in) that fall outside the extent of our confirmation. Further research is expected to investigate and bar such conceivable outcomes. Then again, there is most likely a lot of \slack" in our outcome, and in this manner the bound could be made snugger and/or reached out to more general assaults. It is very conceivable that the genuine lower bound on the quantity of dynamic AND entryways can be much bigger.

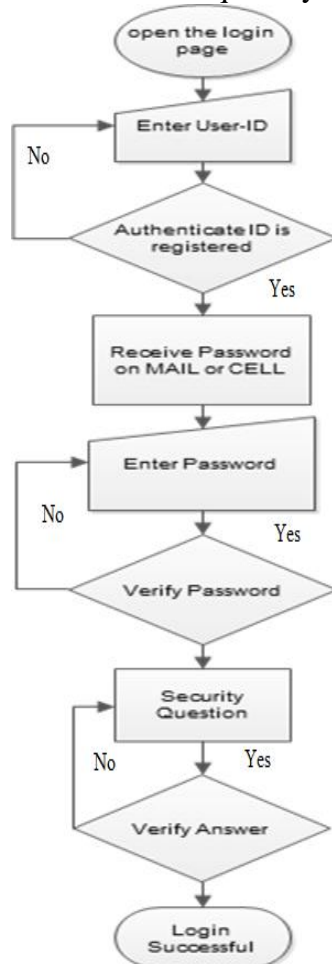


Figure 6: Flowchart of The Proposed Model

Here is the rundown of the reasons: Some of the differential way weight examples may not compare to any legitimate differential way of MD6. The s-round differential way weight design that our system found is not iterative, thus it can't be connected to yield an example for rounds. It didn't number the AND doors over the limit of two back to back rounds. It would likewise be intriguing to investigate how these limits may be influenced by different decisions for the tap positions.

MD6: Linear cryptanalysis

Linear cryptanalysis, spearheaded by Matsui is a kind of assault on square figures that from numerous points of view looks like differential cryptanalysis. Without a doubt, the duality between the two has encouraged synchronous security investigation of a piece figure against differential and direct cryptanalysis. Despite the fact that the structure of MD6 does not permits direct parallel investigation as for the two sorts of assaults, the general methodology and strategy is fundamentally the same. Primary result is a lower bound confirmation demonstrating that any standard direct crypt analytical assaults on keyed MD6 requires no less than 2210 hash info/round sets connected with the obscure key. Subsequently, a key utilization arrangement that obliges the key change after every 2210 messages would thrashing such assaults.

Kerberos

Confounders, Cipher Block Chaining, and Known Plaintext Attacks

Confounders [17] don't puzzle much against known plaintext assaults. A known plaintext assault, rather than a picked plaintext assault, is one in which the individual attempting to discover the encryption key is not permitted to pick the plaintext that is encoded, yet is given a duplicate of some plaintext and the related ciphertext.

Consider an endeavor to find a specific server's key, and accept that all messages contain both a confounder and a checksum, and that figure piece affixing is utilized with the picked encryption plan. It may not be conceivable to focus the plaintext estimation of the first piece of the information, that is, the part that contains the confounder and the checksum. Be that as it may, move along a couple of squares.

The Confounder as our Chosen Plaintext

By definition the confounder should frustrate us. In some sense, however, it might actually have the capacity to pick its esteem. Accept that it has broken the PRNG, and that the confounder and our block size are the same little size. Along these lines, for any given ticket, before asking for the ticket recognize what the confounder will be. Additionally expect that there is some square after the confounder piece that can be pick the plaintext estimation of (call this B), and that all the fields between the confounder and our picked field continue as before for every ticket. Pick a plaintext esteem that you need to know the scrambled estimation of. Stay informed regarding the accompanying data for a few tickets: the confounder quality, and the encoded estimation of the square promptly going before your picked plaintext piece (call this worth E). At the point when anticipated that the following confounder will be one of

the ones that have been seen some time recently, XOR your picked plaintext with E, and supplement this xor esteem into your ticket ask for as your worth for B. At the point when the CBC encryption is carried out, B will be xor'ed with the E once more, and hence your picked content will be the worth scrambled. By the birthday mystery, since the confounder is little it shouldn't take too much sooner than it has a copy entrance which permits us to scramble our picked plaintext.

Differential Cryptanalysis and Breaking Kerberos

Biham and Shamir's [18] assault on DES is of the "picked plaintext" mixed bag. In this way the figure square affixing of the current working draft's suggestion viably keep its utilization in breaking Kerberos. It is imperative that the figure piece binding stay a piece of all encryption routines that utilization DES to avoid Biham and Shamir's strategy to be usable. Under a few circumstances, even with figure piece tying, Biham and Shamir's technique would be pertinent. The thinking is as per the following: assume that the scrambled estimation of the square promptly going before first full piece of your approval information field before it ask for the ticket. Anyway, the encoded rendition of that square, then it could pick the plaintext form of full approval information field piece so that when it is XORed with the past encoded square the outcome is the "plaintext" esteem that it really wish to encrypt. It would be valid for the situation where there were no confounder, checksum, and if the key field containing the session key were moved after the approval information field. For this situation, smart ticket appeals are made, none of the information going before the approval information field would ever change, and subsequently having gotten one ticket you would know the consistent worth for that encoded piece. Assume the measure of a key is genuinely little, thus by the birthday Catch 22 you don't need to hold up long to "haphazardly" pick the same worth for the key once more. At that point regardless of the fact that the key field is before the approval information field, encode the instant message a few times until the scrambled estimation of the square containing the key is the same as in first endeavor, and afterward take a gander at the subsequent ciphertext of approval information field piece. Additionally it could hold up for a little confounder to return, and apply the strategy. Clearly, the above contention is just as pertinent to any system for encryption that can be broken utilizing a picked plaintext assault.

One Time Password

The masquerade attack consists of aborting phase and masquerading phase.

Impact of this attack

The significance of this attack [17] is that the client and the server can't identify the aggressor's masquerade. This masquerade assault does not adjust any component of the chain. Also, the component of the chain that the client figures for the verification to the server relies on upon the counter C_i from the server, not upon a synchronized worth or occasion. In this way, the aggressor can self-assertively prematurely end a client's login stage and afterward disguise the client without location.

One-time watchword confirmation scheme [16] is still helpless against a masquerade assault, however guaranteed that their change on the Yeh-Shen-Hwang plan can withstand the stolen-verifier assault. By prematurely ending one general login session and recording those transmitted messages, an assailant can effectively take on the appearance of the exploited person client to login without location.

Conclusion & Future Scope

This paper proposed a fresher security structure for distributed computing environment which incorporates OTP secret key generator to validate clients, MD-6 hashing for data concealing and Kerberos for Ticket generation. This model guarantees security for entire distributed computing structure.

Here, execution time is not in this way high on the grounds that usage of every calculation is carried out in diverse servers. In the proposed framework, a gatecrasher can't undoubtedly get data and transfer the documents in light of the fact that he needs to take control over all the servers, which is very troublesome. The model, however it is created in a cloud environment, individual servers' operation has got need here. In this way, choice taking is simple for every server, in the same way as validate client, and offer access to a record and so on.

In future, work will be carried out in guaranteeing secure correspondence framework in the middle of clients and framework, client to client.

References

- [1] Ronald L. Rivest ,Benjamin Agre,Daniel V. Bailey, Christopher Crutcheld,YevgeniyDodis, Kermin Elliott Fleming, Asif Khan Jayant Krishnamurthy, Yuncheng Lin Leo, Reyzin Emily Shen, Jim Sukha. Drew Sutherland EranTromer, Yiqun Lisa Yin. The MD6 hash function-A proposal to NIST for SHA-3, October 27, 2008.
- [2] KawserWazed Nafil, TonnyShekhaKar, Sayed AnisulHoque, Dr. M. M. A Hashem4. A Newer User Authentication, File encryption and Distributed Server Based Cloud Computing security architecture.
- [3] ShikhaChoksi. Comparative Study on Authentication Schemes for Cloud Computing, 2014 IJEDR | Volume 2, Issue 2 | ISSN: 2321-9939.
- [4] Ronald L. Rivest, Benjamin Agre, Daniel V. Bailey, Christopher Crutch_eld,YevgeniyDodis, Kermin Elliott Fleming, Asif Khan, Jayant Krishnamurthy,Yuncheng Lin, Leo Reyzin, Emily Shen, Jim Sukha, Drew Sutherland, EranTromer, Yiqun Lisa Yin. The MD6 hash function - A proposal to NIST for SHA-3. Submission to NIST <http://groups.csail.mit.edu/cis/md6/docs/2009-04-15-md6-report.pdf>, 2009.
- [5] Ronald L. Rivest. OFFICIAL COMMENT: MD6. NIST mailing listhttp://groups.csail.mit.edu/cis/md6/OFFICIAL_COMMENT_MD6_2009-07-01.txt), 2009.

- [6] R. L. Rivest. "The md6 hash function, a proposal to nist for sha-3," October 2008. [http://groups.csail.mit.edu/cis/md6/submitted-2008-10-27/Supporting Documentation/md6 report.pdf](http://groups.csail.mit.edu/cis/md6/submitted-2008-10-27/Supporting%20Documentation/md6%20report.pdf)
- [7] C. Y. Crutchfield. "Security proofs for the md6 hash function mode of operation," Master Thesis, Massachusetts Institute of Technology, June 2008. [http://groups.csail.mit.edu/cis/md6/docs/2008-06-crutchfieldms thesis.pdf](http://groups.csail.mit.edu/cis/md6/docs/2008-06-crutchfieldms%20thesis.pdf)
- [8] John Pereless. What is cloud computing. Definition from WhatIs.com. [http://johnpereless.wordpress.com/2014/09/15/cloud-computing-by-john pereless/](http://johnpereless.wordpress.com/2014/09/15/cloud-computing-by-john%20pereless/). Margaret Rouse. WhatIs.com. <http://searchcloudcomputing.techtarget.com/definition/cloud-computing>.
- [9] Roman Mashchak. A Technology Business Guide to Success. [http://www.scribd.com/doc/45798823/A-Technology-Business-Guide-to Success](http://www.scribd.com/doc/45798823/A-Technology-Business-Guide-to%20Success)
- [10] B. Clifford Neuman and Theodore Ts'o. Kerberos: An Authentication Service for Computer Networks. Copyright © 1994 Institute of Electrical and Electronics Engineers. Reprinted, with permission, from IEEE Communications Magazine, Volume 32, Number 9, pages 33-38, September 1994. USC/ISI Technical Report number ISI/RS-94-399.
- [11] EOTP – Static Key Transfer. Defuse.ca (2012-07-13). Retrieved on 2012-12-21.
- [12] Barkan, Elad; Eli Biham, Nathan Keller (2003). "Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication". Cryptome 2003: 60016.
- [13] Barkan, Elad; Eli Biham, Nathan Keller. "Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication by Barkan and Biham of Technion".
- [14] Gueneysu, Tim, Timo Kasper, Martin Novotný, Christof Paar, Andy Rupp (2008). "Cryptanalysis with COPACOBANA". Transactions on Computers Nov. 2008 57: 1498–1513.
- [15] Chun-Li Lin, Ching-Po Hung, Department of Computer Science and Information Engineering Shu-Te University, cclin@mail.stu.edu.tw, 118760@mail.csc.com.tw.
- [16] Jennifer Kay, September 1995, CMU-CS-95-115 School of Computer Science /Carnegie Mellon University /Pittsburgh, Pennsylvania 15213-3890.
- [17] [Biham 90] E. Biham and A. Shamir, *Differential Cryptanalysis of DES-like Cryptosystems*, Technical Report CS90-16, Department of Applied Mathematics and Computer Science, The Weizmann Institute of Science, July 1990.