Certain Investigations of Security Algorithms and its Vulnerabilities for Wireless Networks

M.Malathi,
Research Scholar,
Bharathiar University,
Coimbatore,
Tamilnadu, India

G.M.Kadhar Nawaz Ph.D.,
Research Guide
Director , Dept. Of Computer Applications,
Sona College of Technology , Salem -5
Tamilnadu.

ABSTRACT

Wireless networks is a collection of nodes that communicate in a open-ended manner. The communication among the nodes allow users to communicate from different places in a boundary. However , this forces lots of challenges for the users as the packets pass in a wireless medium and can overlap with other transmissions. This has been the key factor for this survey presented in this paper. The security algorithms exist separately and can be combined with network model leading to different security protocol standards. This paper provides research direction for security algorithms considering the behavioral aspects of the users namely keyboard dynamics. A complete survey has been presented considering the keyboard dynamics analyzing how it can be combined with the security algorithms. A broad variety of applications have been analyzed and the suggestions have been presented. A complete list of performance parameters also have been listed at the end of the survey. The paper provides a list of directions for using neural approaches for authenticating users with keyboard dynamics.

Keywords : Feature vector ,keyboard dynamics ,keystroke, keying patterns, keylength

I. INTRODUCTION

In recent years, security has become an important issue in web applications and several technologies used to meet this. The application of these technologies is to keep secrets safe and secure, but there are pitfalls involved with utilizing them. This project focuses on one of the vital components used in various security related technologies namely "randomness". This component is by nature, complex and easily misunderstood[11]. One may say that randomness plays a "key" part in most cryptosystems today.

A large encryption key is said to be strong, if it is unknowable to a potential attacker. This requires the input of good random numbers during key generation. If the inputs to the key generation are not random, an attacker will be able to exploit the statistical bias. Good randomness, unfortunately, is difficult to produce for modern computers. As already mentioned computers are calculating machines which perform predefined operations according to predefined scripts, called programs[9]. Nothing about a computer is random. Computing is 100% deterministic albeit complex and sometimes opaque to the human observer. To compensate for this shortcoming, random number generators accept what is referred to as a seed. The seed initializes the internal state of the random number generator and thus sets a starting point. Thereafter a complex mathematical sequence is applied to produce statistically pattern free output. If the starting point, or seed, to the mathematical sequence is unknowable, the random number generator can be said to be "truly random"[12].

Random numbers are numbers that occur in a sequence such that two conditions are met:

- **Uniform distribution:** The distribution of the numbers in the sequence should be uniform; that is, the frequency of occurrence of each of the numbers should be approximately same.
- **Independence:** No one value in the sequence can be inferred from the others.

Although there are well-defined tests for determining that a sequence of numbers matches a particular distribution, such as the uniform distribution, there is no such test to "prove "independence. Rather, a number of tests can be applied to demonstrate that sequence does not exhibit independence[10]. The general strategy is to apply a number of such tests until the confidence that independence exists is sufficiently strong.

A number of network security algorithms based on cryptography make use of random numbers. For example,

- **Reciprocal authentication schemes**. In the both of these key distribution scenarios, nonces are used for handshaking to prevent replay attacks. The use of random numbers for the nonces frustrates opponents' efforts to determine or guess the nonce.
- **Session key generation**, whether done by a key distribution center or by one of the principals.

Generation of keys for the RSA public-key encryption algorithm.

The concept of random numbers belong to Group Theory. Generally there are two types of random Numbers: True Random Numbers and Pseudo Random Numbers.

II. RELATED WORK

The pseudo random approach can be done using entropy harvesting approach. A clear understanding on the need on Pseudo random numbers is explored. The problem areas related to security breaches in Linux is explored. Pseudo random generator designed has value n larger than k and is very harder to predict.

The approach for generating entropy using random bits is also an alternative one. There is more focus for generating entropy value based on the entropy rate. The fundamental approach is to exploit the symmetric of the source. The fixed and variable random number generator is considered in detail for selection.

The practical problems in generating the random numbers in software could also be analyzed. There are possibilities to use /dev/random and dev/random for analysis the random numbers generated. The two major aspects of PRNG are security and generation of data in bulk level. OpenSSL also provides mechanisms for handling the security at a better level. PRNG along with entropy harvesting is considered in this work.

A random number generator based on the curve points is also a better approach. The developed approach was towards saving hardware and software components. In general the bit length of a random number is dependent on the field size. The random number generation may involve more than one cycle. By changing the seed and the initial point different bit sequence can be generated. Point P can be combined with ECC but P cannot be a point as part of the ECC. This provides basis that points in the ECC are public keys. The given bit sequence input is processed and the ECC algorithm generates the key based on the curve fitting points.

Author /Year	**Methodology**	**Merits**	**Demerits**	**Application**	**Comments**
Benzamin Arazi/1989	A secret communication between key and image	Two level security Computationally hard to break Different Reference grids for pseudo random number generation	Does not consider the platform issues	Facial recognition	Could be extended with platform implementation and can be studied.
T. Kiesler ,1992	Akl-Taylor-MacKinnon-Meijer scheme is used	Master Key generation scheme is proposed Can be applied for Group key generation	Works upto limited depth in key generation	Social network	Can be applied in real time environment Could be implemented with agents
R. J. Anderson	Trapdoor mechanism	Design Trapdoor and implementation trapdoor is proposed	RSA computationally issues not analyzed	Banking Systems	Threats could be analyzed and integrated to the approach.
Vlad. M. Sidelnikov	A modified cyclic group analysis	A Group generation is done based on cyclic group analysis	Implementation part is not completed	Criticical Alarm System	Key behavior could be integrated.
Raymond W. Woo	Derangement Permutation method	Scarmbles keys in a frequency domain representation system	Has not been tested for sound with noised	Voice Authentication System	Could be extended with more training and results can be presented.
R. Poovendran	Trust Authentication Model	Verifiable Secret Sharing techniques	Has not been tested for real time data	Transaction oriented applications	Can be integrated with different keyboards.
Stefan Wolf	Secret Key aggrement model	Special scenarios are analyzed where X, Y , and Z are generated by sending a binary random variable R, for example a signal broadcast by a	The robustness of the mathematical model proposed can be tested	Banking application	Can be tested with different platforms

		satellite			
Imre Csiszár	Authentication using multiple terminal source	Multiple entropy sources has been modelled	Does not integrate with any key generation scheme	System authentication in a hiereachial network	A Trust model can be generated
Richard J. Hughes	Quantum Cryptography	Quantum cryptography for 0.5Km is applied.	Not applicable for distributed users	Banking Application	Can be tested with wide spread users
Bin Sun	Group Oriented Trusted Model	Butterfly Key Tree Structure is proposed Dynamic member key assignment is provided.	Intruder analysis is not handled.	Third party applications.	Can be modeled using agents.
Daniel Gottesman	Two Way entanglement Purification Model	Bit Error Tolerance upto 18.9% Applicable for long distance communication	Has not been tested for real time platforms	Inventory applications	Could be enhanced with behavioural analysis.

(i) Authentication Protocols

The **challenge-response protocol** is a fundamental tool of secure authentication. This is a process that verifies an identity by requiring correct authentication information to be provided in response to an unpredictable challenge.The challenge is usually a random number5, and the response is related to this number. Use of this protocol prevents an attacker from replaying a previous authentication response. Below, we describe basic protocols for passwords, tokens, and biometrics. This is to show how each authenticator can participate in a challenge-response protocol and how the authenticator information is stored at the host. Although the protocol we describe in Case 1 is the basis for such widely used password protocols as Unix and Windows NT and 2000 login, the actual protocols are generally more complex[17].

Case 1: Password Protocol – The basic password challenge-response protocol is initiated when a user sends user identification, *U*, to the host in step 1. (See Figure 2.2.1.) In step 2, the host returns a random number, *r*, that will identify the session, a hash function, *h()*, and a challenge function, *f()*. In step 3, the user returns the response, comprised of the result of the function involving the hash of a submitted password, *h(P'),* and the submitted random number, *r'*. In step 4, authentication is granted if this result is equivalent to the result of the function with random number and the hash of the true user password, *h(P(U))*; otherwise it is not granted. Note that the user password, *P(U)*, is not stored in plaintext on the host; instead it is hashed to form *h(P(U))* to avoid theft at the host[18].

Case 2: Token Protocol – In the basic token authentication protocol, the token either stores a static passcode or generates a one-time passcode. This is similar to the password protocol, however instead of a potentially weak password, a long and random passcode is first hashed, *h(W'))*, combined with the random number challenge, and then transmitted as the response to the host. The user accesses the passcode from token storage with a password, *P'*, but that password is used only between the user and the user-held token. The user passcode can be stored in hashed form at the host, *h(W(U))*, or it can be generated for one-time passcodes. Authentication of the password at the token can be done similarly to Case 1.

The following two cases involve biometric matching. Case 3 pertains to a stable biometric signal or to an alterable biometric signal that does not take advantage of its alterability to engage in a challenge-response protocol. Case 4 describes a challengeresponse protocol that can only involve alterable biometrics[19].

Case 3: Stable Biometric Protocol – This is a basic challenge-response protocol for a stable biometric that is matched at the host. (See Figure 2.2.3.) A biometric, *B'*, is captured and processed on a biometric device at the client to obtain a biometric template, *BT'*. This template is combined with the random number challenge, *r'*, then encrypted, *E()*, and returned as the response to be matched at the host[20]. In Figure 2.2.3, we also show a rudimentary procedure for authentication of the capture device where the device returns its identification, *D'*, that is compared with a list of registered devices at the host database, *{D}*.

The basic challenge-response protocol for a stable biometric that is matched at the client is similar to that matched at the host. The distinction is that a biometric is captured, processed to a template, *BT'*, and matched to yield a yes/no match result, *BM'*, all at the client. The information is transmitted to the host, which determines authentication depending on a correct match and the legitimacy of the biometric device. The host contains no biometric information; instead the biometric template is stored at the client.

Case 4: Alterable Biometric Protocol – This is a basic challenge-response protocol for an alterable biometric signal that is matched at the host. (See Figure 2.2.4.) One difference from the stable

biometric signal is that we can now involve the actual biometric in challenge-response, whereas we could not before[21]. To do this, a challenge, *x*, is sent from the host to the client. This challenge is a random sequence of numbers, characters, or
words. This is much shorter than the random number, *r*, because the user will have to vocalize it (speaker verification), type it (keyboard dynamics verification), or write it (handwriting verification) to yield the biometric signal, *BS'(x')*. This response is returned to the host, where processing is done to extract *x'* and *B'*. The recognized *x'* is compared with the challenge originally sent, *x*. The biometric, *B'*, is compared with that in the database corresponding to the user, *B(U)*. If *B'* matches *B(U)* and if *r'* matches *r*, then
authentication is successful. Note a difference here from the stable biometric protocol is that the capture device need not be machine-authenticated. There is no need to do this here since the challenge-response protocol defends against replay and forgery, and matching is performed at the host.

The basic challenge-response protocol for an alterable biometric signal that is matched atthe client is similar to that matched at the host. The distinction is that a biometric is captured, processed to a template, *BT'*, and matched to yield a yes/no match result, *BM'*,all at the client. The result is sent to the host along with a device identifier to verify that it is registered and unmodified. As compared with host matching, this protocol saves transmission bandwidth and template storage space at the host, at the cost of a more powerful and trustworthy device at the client.

	Client	Transmission	Host
1	U,user.	U-->	
2		<--{r,h(),f()}	r,random num h(),f(),functions.
3	P',passsword r',random num	f(r',h(p'))-->	
4		<--yes/no	If f(r',h(P'))=f(r,h(P(U))) then yes:else no

2.1 Basic challenge-response protocol for a password(case 1)

	Client	Transmission	Host
1	U,user.	U-->	
2		<--{r,h(),f()}	r,random num h(),f(),functions.
3	P'-->W' password to passcode via token; r',random num	f(r',h(W'))-->	
4		<--yes/no	If f(r',h(W')=f(r,h(w;(U))) then yes:else no

2.2 Basic Challenge-response protocol for a token(case 2)

	Client	Transmission	Host
1	U,user.	U-->	
2		<--{r,E()}	r,random num E(),function
3	B'-->BT',biometric; D',biometric device; r',random num	E(r',D',BT')-->	E(r',D',BT') = {r',D',BT'}
4		<--yes/no	If r'= r and D'e {D} and BT' = BT(U) then yes;else no

2.3 Basic Challenge-response protocol for stable biometric.(case3)

	Client	Transmission	Host
1	U,user.	U-->	
2		<--{r,x,E()}	r,random num; x,random seq,challenge; E() functions
3	B',x'-->BS'(x'); r',random num	E(r',BS'(x'))-->	E(r',BS'(x')) = {r',BS'(x')}, BS'(x')-->BT'(x') = f(B',x'), Recognize x'from BS'(x') Extract B'from BT'(x')
4		<--yes/no	If r' = r, and x' = x, and B' = B(U), then yes;else no

2.4 Basic Challenge-response protocol for alterable biometric(case 4)

3. PERFORMANCE MEASURES

The different performance measures that can be used for authentication is presented below:

(i) Bayesian Classification:

Statistical method is a method for feature matching. This method will verify the user based on statistical data such as mean and standard deviation.

$$Statscore = \frac{1}{n}\sum_{i=1}^{n} e^{-|ti-\mu i|/\sigma i}$$

Where ti is the *i*-th test feature, μi and σi are mean and standard deviation of the reference template,

respectively.Constant *e* is set as 2.71828.

(ii) Euclidean Distance Measure:

Euclidean distance measure: "similarity" is based on the Euclidean distance between the pattern vectors.It is an simple approach for comparing the typing patterns of training sample of user and testing pattern.

The consistency of the user's typing habits is determined by positive value implies time reduction (faster in pressing) and negative value implies time addition (slower in pressing) between two sequences of keystrokes.

(iii) MCMC-MarkovChain Monte Carlo

Method for identifying the authenticated user

$$Final_{score}=\sum(w_i*score^i).$$

Where $\Sigma wi=1$ and *scorei* is the score of *i*-th method used.

Local threshold used in verification system is threshold value between the input data and the model. One way to estimate the value of local threshold is by using the genuine user data, impostor data or a combination of both . If the result of feature **matching score < threshold**, then the user is identified as an impostor, otherwise the user is identified as an actual user.
The equation used to determine the local threshold value is

$$\theta = \mu user - \alpha , \sigma user$$

Where θ - local threshold,
μuser - mean score from user enrollment,
α- a constant factor obtained from the experiment.

The Keystroke Dynamics is relatively needs only a keyboard and software for authentication, different of the others biometrics techniques that possess one high cost of the captation devices and analysis of the necessary data in the authentication, and can also be used with or without the knowledge of the person.

Some features can be extracted of the keystroke rhythm as:

1. The time that a key is pressed (keystroke duration),
2. The time between successive keys (keystroke latency),
3. Speed of the keystroke,
4. Placement of the fingers and
5. Pressure that the person applies when pressing a key (pressure keystroke) .

4. CONCLUSION

The paper has made a complete survey of different techniques available for authentication. This has led to the following research directions:

i. Enhancement with Visual Cryptography with images
ii. Third party authentication based on statistical data
iii. User authentication based on keyboard category like Indian,Chinese keyboard
iv. Generation of test data set using neural networks and then authenticating the users.
v. Identification of outliners for effectively identifying users.
vi. Classifying users based on the type of behavior based on different classifiers.
vii. Integrating the schemes that are mathematically derived to be part of the network standard.

5. REFERENCES

[1] Benzamin Arazi ,ITSHAK DEINSTEIN,ODED KAFRI,"Intiution Perception And Secure Communication", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, VOL. 19, NO. 5, SEPTEMBER/OCTOBER 1989

[2] T. Kiesler, L. Harn, " Cryptographic Master-Key-Generation Scheme And Its Application To Public Key Distribution", IEE PROCEEDINGS-E, Vol. 139, No. 3, MAY 1992

[3] R. J. Anderson," PRACTICAL RSA TRAPDOOR", ELECTRONICS LETTERS 27th Mav 1993 Vol. 29 No. 11

[4] Vlad. M. Sidelnikov , 'Exponentiat Ion-Based Key Generation Using Noncommutat Ive Groups",IEEE,1994

[5] Raymond W. Woo And Cyril Leung," A New Key Generation Method For Frequency-Domain Speech Scramblers", IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. 45, NO. 7, JULY 1997.

[6] R. Poovendran, M. S. Corson, J. S. Baras," A DISTRIBUTED SHARED KEY GENERATION PROCEDURE USING FRACTIONAL KEYS", IEEE,1998.

[7] G. J . Simmons, Llainn Troduction To Shared Secret And/Or Shared Control Schemes And Their Applications", G. J. Simmons, Editor, Contemporary Cryptology: The Science Of Information Integrity, 441-497, IEEE Press, 1992.

[8] Ueli M. Maurer, Stefan Wolf," Unconditionally Secure Key Agreement And The Intrinsic Conditional Information", IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 45, NO. 2, MARCH 1999

[9] W. Diffie And M. Hellman. "New Directions In Cryptography",IEEE Trans. Inform. Theory IT-22, 6 Nov. 1976, 644-654

[10] D. Bressoud, "Factorization And Primality Testing, Berlin:Springer-Verlag, 1989

[11] A. Abdul-Rahman, S. Hailes. "A Distributed Trust Model",New Security Paradigms Workshop, Great Landale, UK.1997.

[12] R. Yahalom, B. Klein. T. Beth. "Trust Relationships In Secure Systems - A Distributed Authentication Perspective.Proceedings Of The IEEE Symposium On Research In Security And Privacy, 1993.

[13] Imre Csiszár," Common Randomness And Secret Key Generation With A Helper", IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 46, NO. 2, MARCH 2000

[14] Richard J. Hughes, William T. Buttler, Paul G. Kwiat, Steve K. Lamoreaux,University Of Califomia," Quantum Cryptography For Secure Satellite Communications",IEEE,2000.

[15] Bin Sun,Wade Trappe, Yan Sun And K. J. Ray Liu," A Time-Efficient Contributory Key Agreement Scheme For Secure Group Communications",IEEE,2002.

[16] Daniel Gottesman And Hoi-Kwong Lo," Proof Of Security Of Quantum Key Distribution With Two-Way Classical Communications", IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 49, NO. 2, FEBRUARY 2003.

[17] B. Martin, P. Isokoski, F. Jayet, T. Schang, "Performance Of Finger-Operated Soft Keyboard With And Without Offset Zoom On The Pressed Key," Proceedings Of The 6th International Conference On Mobile Technology, Application & Systems, Nice, France, 2009.

[18] J. Goodman, G. Venolia, K. Steury, C. Parker, "Language Modeling For Soft Keyboards," Proceedings Of The 7th International Conference On Intelligent User Interfaces, January 13-16, 2002, San Francisco, California, USA.

[19] P. Isokoski, "Performance Of Menu-Augmented Soft Keyboards," Proceedings Of The SIGCHI Conference On Human Factors In Computing Systems, P.423-430, April 24-29, 2004, Vienna, Austria.

[20] I. S. Mackenzie, S. X. Zhang, And R. W. Soukoreff, "Text Entry Using Soft Keyboards," Behaviour & Information Technology, 18, 235--244, 1999.

[21] Pekka Parhi , Amy K. Karlson , Benjamin B. Bederson, "Target Size Study For One-Handed Thumb Use On Small Touchscreen Devices," Proceedings Of The 8th Conference On Human Computer Interaction With Mobile Devices And Services, September 12-15, 2006, Helsinki, Finland.