Mobile Networks and Applications (2019) 24:1249–1254
https://doi.org/10.1007/s11036-019-01263-1

Packet Encryption for Securing Real-Time Mobile Cloud Applications

Ajay D M[1] · Umamaheswari E[1]

Published online: 29 April 2019

Abstract

The evolution of smart devices has brought the most confidential data to mobile applications and cloud computing with its processing power is serving the huge processing requirements of these applications. However, these advancements has brought in serious data security concerns, as there is constant shuttling of data from devices to cloud and backforth, and encryption is the most commonly used technique for securing data in transmission. Every encryption technique is breakable, but its efficiency is calculated with the time it can withstand an attack. But in today's world, as cloud computing comes with almost unlimited computing resources, the present day encryption techniques might become inadequate for securing confidential data in transmission and after transmission. Existing encryption techniques, encrypt the whole data packets in a session using the same session key. If a third party can decrypt a single packet during or after transmission, all the packets in that session can be compromised. This work proposes a packet encryption scheme, where a packet key once used will never be used throughout the session, with minimal latency and maximum security for real-time mobile cloud applications.

Keywords Real-time Mobile Applications · Data Security · Packet Encryption · Mobile Cloud Computing

1 Introduction

The advancements in mobile phones has brought in the crucial data applications into mobile devices. Most of the desktop applications like banking applications, storage applications, mailing applications, commerce applications, social networking applications, etc. are available in mobile devices. Mobile applications with the proliferation of cloud computing has propelled into a new class of applications called Mobile Cloud Computing (MCC) applications [1]. In contrast to the traditional applications, MCC applications comes with unlimited storage and computing resources, with services that are task-oriented. MCC applications has moved the mobile phone's computation power and storage to the cloud [2]. MCC applications include multimedia sharing, mobile learning, mobile sensing, mobile healthcare, mobile gaming, location-based mobile services etc.

Cloud Computing has enabled to accommodate large digital images, music, video files into the smart phones. A major portion of processing of MCC applications is handled in the cloud, this improves the overall user experiences of smart phone with faster and efficient processing, synching, and extended battery life etc. [3]. Cloud has also provided abundant storage space for the smart phone users and has improved the data synching capabilities between different devices of the user.

However, these advancements have brought in serious data security concerns, as there is a constant shuttling of data from the mobile devices to the cloud and back forth. This makes data security in MCC one of the major challenge. Data in MCC should be secured in the end-user device side, cloud side and during transmit [4]. Using MCC applications comes with both the security concerns related to cloud and attacks at End-User Mobile devices [5]. Some of the data concerns in cloud are Risk of data theft, Handling of encryption and decryption keys, Violation of privacy rights, etc. It is also important to protect the data at the end-user devices from data theft, malware, and virus attacks.

Encryption is the most commonly used technique to secure the data, the existing techniques encrypt all the data packets in a session with the same session key. One of the biggest drawbacks of all the existing techniques which uses a session key is that, If a Third Party (TP) succeeds in decrypting any one packet in the session, the TP will be

✉ Ajay D M
dm.ajay2015@vit.ac.in

Umamaheswari E
umamaheswari.e@vit.ac.in

[1] School of Computing Science and Engineering, VIT Chennai, Chennai, India

able to decrypt all packets in that session. This work proposes a packet encryption technique, where a packet key once used will never be used throughout the session, with minimal latency and maximum security. [6–8] proposed a Selective Packet key encryption scheme, substituting the session key encryption technique. In Selective Packet key encryption scheme the Diffie-Hellam (D-H) key agreement is repeated every time a packet is sent. D-H allows two parties to share a secret key without sending the original key [9]. Section 2, discusses some of the related works for the proposed scheme. Section 3, discusses the proposed packet key encryption scheme in detail.

2 Existing Methodologies

The existing methodologies or protocols for mobile data security include:

- Secure Socket Layer (SSL), Transport Layer Security (TLS) etc., both SSL and TSL provides web traffic security.
- Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), Wireless Application Protocol 2 (WAP) etc., are used to provide wireless network security.
- Virtual Private Network (VPN) technologies like Internet Protocol Security (IPsec), Point-to-Point Tunneling Protocol (PPTP), Layer 2 Tunneling Protocol (L2TP) etc., are used to provide network security.

SSL is a security standard proposed by the Internet Engineering Task Force (IETF) for establishing a secure encrypted link between a web browser and server., and SSL v3.0 is a widely implemented protocol and is the basis for TLS [10]. SSL technology ensures that all the data transmitted in the created link remains encrypted, this way SSL ensures that all the data passing between the web browser and server remain private and integral. TLS is the upgraded version of SSL v3.0 proposed by IETF. TLS is more secure and efficient than SSL in message authentication, key material generation and support for cipher suites. TLS is comprised of two layers TLS Record Protocol, TLS Handshake Protocol. TLS Record Protocol provides connection security and TLS Handshake Protocol enables authentication and negotiation of cryptographic keys and encryption algorithms between the client and server [11].

WEP protocol specified in the IEEE wireless standard 802.11b, has been developed in order to provide security and privacy in Wireless Local Area Network (WLAN). WEP establishes protection by encrypting the data transmitted over the WLAN. Data encryption scheme is implemented by combining user and system generated key values. WEP originally supported 64 bits of key length, which includes encryption keys of 40, plus 24 additional bits of system generated data. In WEP an attacker can easily forge an authentication message, which facilitates him to pretend like a legitimate user and steal critical information. WEP is vulnerable against forgery and replay attacks [12]. WPA officially replaced WEP in 2004, after several security researchers discovered flaws in the WEP design. WPA provides security for computing devices equipped with wireless internet connections. WPA provides more efficient data encryption and user authentication than WEP. Temporal Key Integrity Protocol (TKIP) is used in WPAs encryption method which is a wrapper around the existing WEP encryption. TKIP uses 128 bit key and RC4 algorithm for encryption. The main advantage of TKIP is that, key is modified for each and every packet. The key is generated by combining the base key, transmitting station MAC address, and the packet serial number [12]. The authentication is provided by RADIUS, a central authentication server based on EAP – Extensible Authentication Protocol (EAP) and 802.1X. Counter Mode Cipher Block Chaining Message Authentication Code Protocol (CCMP) is an encryption protocol based on the Obligatory Advanced Encryption Standard Algorithm is used by WPA2. Authenticity of Messages and integrity verification is provided by this obligatory Advanced Encryption Standard Algorithm which is more reliable and stronger the original TKIP protocol in WPA [13, 14]. Each packet possess a 48-bit serial number that gets incremented every time a new packet is transmitted. This serial number is used as the initialization vector and also as a part of the key that is used to encrypt the packet. This solves collision attack, which occurs when the same key is repeatedly used for different packets. Replay attacks can be reduced by using the packet serial number as the initialization vector, which is hard to decrypt. A new base key is created every time a wireless station associates to an access point.

VPN enables a safe and encrypted connection by creating a tunnel between the source and destination over a less secure network, like the internet. PPTP is a popular networking protocol in windows machines. PPTP implements a VPN connection which is mostly used by corporations to extend their own network through private networks over the public internet. Using PPTP, users remotely can access their corporate networks securely using windows platforms or Point to Point Protocol devices. PPTP uses PPP for authentication and encryption of data packets [15, 16]. PPTP uses Transmission Control Protocol (TCP) through Generic Routing Encapsulation (GPC) to facilitate VPN connection. PPTP uses a 128-bit encryption for securing data. L2TP is an extension of PPTP protocol used by Internet Service Provider (ISP) to enable VPN [17, 18]. L2TP is a combination of PPTP (Microsoft) and L2F Layer 2 Forwarding (Cisco) protocols. L2TP consists of two primary physical elements, the L2TP Access Concentrator (LAC) and L2TP Network Server (LNS) [19]. Control messages and Data messages are types of messages in L2TP.

IPsec protocol is the most commonly used VPN for enabling secure Internet communications. IPsec can operate in tunnel mode and transport mode [20, 21]. In the

Table 1 Comparison results implementing different encryption algorithms in IPsec

Encryption Type	Average Round Trip time(ms) for different packet sizes			
	84bytes	256bytes	512bytes	1024bytes
DES	39.91	50.62	45.21	52.42
3DES	47.89	50.02	48.03	42.54
AES	49.63	49.01	42.37	50.21

Transport mode the data packet messages alone are encrypted, and in the Tunnel mode the complete packet is encrypted. Authentication Header (AH) provides authentication and integrity to the datagrams and Encapsulating Security Payload (ESP) provides encryption, data origin authentication, data integrity to the datagrams [22]. ESP performs encryption at IP packet layer. IPsec supports a variety of encryption algorithms like DES, 3DES, AES and D-H for establishing a shared secret key. Besides this IPsec supports a variety of hashing algorithms, public key encryption algorithms and application layer protocols like IKE, ISAKMP etc., and Table 1 represents the Comparison results implementing different encryption algorithms in IPsec and Fig. 1 shows the graphical representation of the Comparison results of implementing different encryption algorithms in IPsec.

Aim: To establish a Virtual Private Network using IPsec protocol.
Hardware Configuration: Processor: Intel Core i5, RAM: 8 GB.
Software Configuration: Operating System: Windows 8, Simulation Tool: GNS3 2.0.3, Monitoring Tools: Wireshark, Routers: Cisco7200, Algorithms used: DES, 3DES, AES, Packet Type: UDP, Key Sharing: Diffie-Hellman (D-H) 1024 bits. The following are the implementation results:

The concept of packet encryption was formulated based on the selective packet key encryption scheme by Younchan Jung and Enrique Festijo in [6]. Jung and Festijo [7] states that 17.5% of packet encryption is sufficient to make the data secure during transmission and from after-transmission attacks, by balancing both security and latency. Jung et al. [8] proposes that a packet key scheme with four digit key size is sufficient in providing stronger after-transmission confidentiality than a nine digit key size of session key scheme. The proposed system applies this packet encryption technique for all the packets in a session for MCC applications.

3 Proposed Methodology

This work proposes a packet encryption technique, where a packet key once used will never be used throughout the session, with minimal latency and maximum security for real-time mobile cloud applications. Each packet will be encrypted with different key. Even if a TP successfully decrypts a single packet in a session, the hacker will not be able to decrypt the remaining packets. The TP needs to spend the same amount of time to decrypt each and every packet, which might take years to complete. The encryption algorithm used in this encryption technique might vary based on the size, criticality of the data and the MCC application.

A secure connection is established between the MCC application (*M*) and Cloud server (*S*). In the next step *M* and *S* agree on two global parameters y, α ($\alpha < y$) and α is a primitive root of prime number y. Both *M* and *S* share values of parameters y, α using D-H. In the next step, *M* and *S* will

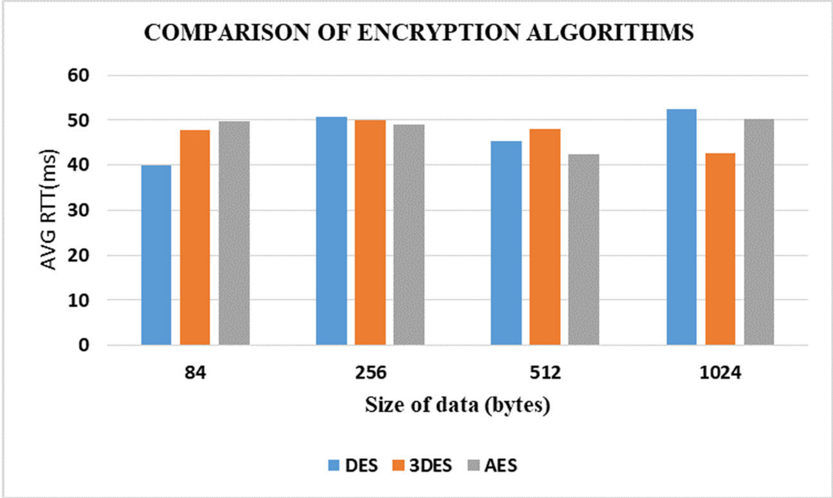

Fig. 1 Shows the Comparison results of different encryption algorithms

select session secret values X_M and X_S respectively and calculate the blind keys $Z_S$and Z_M, using the formula in eq. (1) and (2). Keeping the values X_M and X_S as secret, M and S share the session blind key values Z_M and Z_S (Fig. 2).

If M is the sender, then M generates a packet secret key $X_{M, i}$, using the values y, α and $X_{M, i}$,and calculates the corresponding blind key $Z_{M, i}$ using the equation:

$$Z_{M,i} = \alpha^{X_{M,i}} \; mod \; y \tag{1}$$

If S is the sender, then S generates a packet secret key $X_{S, i}$, using the values y, α and $X_{S, i}$, and calculates the corresponding blind key $Z_{S, i}$ using the equation:

$$Z_{S,i} = \alpha^{X_{S,i}} \; mod \; y \tag{2}$$

Using Z_S, $X_{M, i}$, y and α, M will calculate the packet key $K_{M, i}$, using the equation:

$$K_{M,i} = {Z_S}^{X_{M,i}} \; mod \; y \tag{3}$$

Using Z_M, $X_{S, i}$, y and α, S will calculate the packet key $K_{S, i}$, using the equation:

$$K_{S,i} = {Z_M}^{X_{S,i}} \; mod \; y \tag{4}$$

The packet key $K_{M, i}$ of M is given as an input to the encryption algorithm and then the encrypted payload is generated $E_{PK^{M,i}}(P_{M,i}) = E(K_{M,i}, P_{M,i})$. Likewise, the packet key $K_{S, i}$ of S is given as an input to the encryption algorithm and then the encrypted payload is generated $E_{PK^{S,i}}(P_{S,i}) = E(K_{S,i}, P_{S,i})$. When the encrypted packet reaches s, the packet is searched for the packet blind key $Z_{M, i}$, if the packet blind key is found, s computes the packet key $K_{M, i}$ using $K_{M,i} = {Z_{M,i}}^{X_S} \; mod \; y$. Then the $K_{M, i}$ value is given as an input to the encryption algorithm the encrypted payload is decrypted using $D_{PK^{M,i}}[E_{PK^{M,i}}(P_{M,i}) = E(K_{M,i}, P_{M,i})]$. The same Process happens when s sends an encrypted packet to M, the packet blind key is searched, if found $Z_{S, i}$, packet key is calculated using $K_{S,i} = {Z_{S,i}}^{X_M} \; mod \; y$ and then $K_{S, i}$ given as input to the encryption algorithm and the encrypted payload is decrypted by using $D_{PK^{S,i}}[E_{PK^{S,i}}(P_{S,i}) = E(K_{S,i}, P_{S,i})]$. Figure 2, Shows the process flow in the proposed packet encryption scheme for MCC Applications. The encryption technique might vary based on the size of the data and criticality level of the data. The proposed packet scheme generates a unique key to encrypt each packet i.e., a packet key once used will never be used in the entire session.

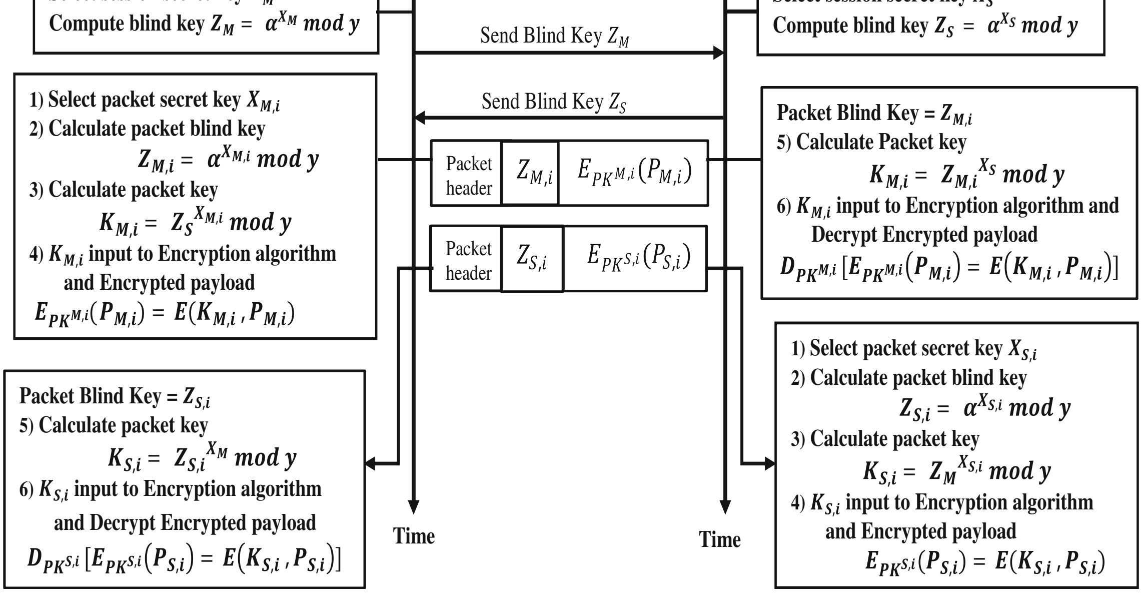
Establishing secure mode connection
Agreement on global parameters y, α
Agreement on global parameters y, α
Send Blind Key Z_M
Send Blind Key Z_S
Select session secret key X_M
Compute blind key $Z_M = \alpha^{X_M} \; mod \; y$
Select session secret key X_S
Compute blind key $Z_S = \alpha^{X_S} \; mod \; y$
1) Select packet secret key $X_{M,i}$
2) Calculate packet blind key
$Z_{M,i} = \alpha^{X_{M,i}} \; mod \; y$
3) Calculate packet key
$K_{M,i} = {Z_S}^{X_{M,i}} \; mod \; y$
4) $K_{M,i}$ input to Encryption algorithm and Encrypted payload
$E_{PK^{M,i}}(P_{M,i}) = E(K_{M,i}, P_{M,i})$
Packet header
$Z_{M,i}$
$E_{PK^{M,i}}(P_{M,i})$
Packet header
$Z_{S,i}$
$E_{PK^{S,i}}(P_{S,i})$
Packet Blind Key = $Z_{M,i}$
5) Calculate Packet key
$K_{M,i} = {Z_{M,i}}^{X_S} \; mod \; y$
6) $K_{M,i}$ input to Encryption algorithm and Decrypt Encrypted payload
$D_{PK^{M,i}}[E_{PK^{M,i}}(P_{M,i}) = E(K_{M,i}, P_{M,i})]$
Packet Blind Key = $Z_{S,i}$
5) Calculate packet key
$K_{S,i} = {Z_{S,i}}^{X_M} \; mod \; y$
6) $K_{S,i}$ input to Encryption algorithm and Decrypt Encrypted payload
$D_{PK^{S,i}}[E_{PK^{S,i}}(P_{S,i}) = E(K_{S,i}, P_{S,i})]$
1) Select packet secret key $X_{S,i}$
2) Calculate packet blind key
$Z_{S,i} = \alpha^{X_{S,i}} \; mod \; y$
3) Calculate packet key
$K_{S,i} = {Z_M}^{X_{S,i}} \; mod \; y$
4) $K_{S,i}$ input to Encryption algorithm and Encrypted payload
$E_{PK^{S,i}}(P_{S,i}) = E(K_{S,i}, P_{S,i})$
Time
Time

Fig. 2 Shows the process flow of the proposed packet encryption scheme for MCC Applications

Table 2 Chi-Squared Test

	y	α	^{X}M	^{Z}M	^{K}M	^{X}S	^{Z}S	^{K}S	Total
$P_{M,\ i}$	4,871,077	4,871,000	2,546,312	4,427,387	2,291,847	2,568,497	1,601,734	2,291,847	25,469,701
$P_{M,\ i}$	4,871,077	4,871,000	3,562,451	991,786	192,401	2,568,497	1,601,734	192,401	18,851,347
$P_{M,\ i}$	4,871,077	4,871,000	1,265,789	3,917,476	1,412,766	2,568,497	1,601,734	1,412,766	21,921,105
$P_{M,\ i}$	4,871,077	4,871,000	4,126,458	4,060,055	2,545,664	2,568,497	1,601,734	2,545,664	27,190,149
$P_{M,\ i}$	4,871,077	4,871,000	2,468,749	4,314,613	3,788,496	2,568,497	1,601,734	3,788,496	28,272,662
$P_{M,\ i}$	4,871,077	4,871,000	1,684,237	4,690,516	812,857	2,568,497	1,601,734	812,857	21,912,775
$P_{M,\ i}$	4,871,077	4,871,000	3,647,215	3,744,473	1,110,982	2,568,497	1,601,734	1,110,982	23,525,960
$P_{M,\ i}$	4,871,077	4,871,000	4,391,657	1,152,414	3,529,951	2,568,497	1,601,734	3,529,951	26,516,281
$P_{M,\ i}$	4,871,077	4,871,000	2,167,342	1,986,350	1,341,980	2,568,497	1,601,734	1,341,980	20,749,960
$P_{M,\ i}$	4,871,077	4,871,000	3,249,617	807,885	3,101,334	2,568,497	1,601,734	3,101,334	24,172,478
Total	48,710,770	48,710,000	29,109,827	30,092,955	20,128,278	25,684,970	16,017,340	20,128,278	238,582,418
								Chi-square:	21,106,520.62
								degrees of freedom:	63
								ρ-value:	0
								Yates' chi-square:	21,106,500.56
								Yates' ρ-value:	0

Latency and Security are major concerns especially for real-time mobile applications, obtaining a balance between these parameters is crucial for proving the efficiency of the proposed packet encryption scheme. Latency in the proposed scheme is caused by the Diffie-Hellman (D-H) key agreement, encryption and decryption for each packet. D-H key agreement is discrete logarithmic computation by nature. For investigating the effects of latency, firstly, one of the existing methodology IPsec protocol has been simulated using GNS3 tool and the results have been published in Section 2: Table 1 and Fig. 1. IPsec uses a single key for an entire session but the proposed scheme uses different keys for each packet in a session. So the average latency caused by packet key exchange for the entire session in the existing methodology is equal to a single packet key exchange in the proposed scheme. According to Younchan Jung and Enrique Festijo in [6], the key size of the packet should be below 8-digit key in order to maintain the tolerable latency of real-time applications. In this proposed scheme, the packet key size is fixed at 7-digit key for maintaining the tolerable latency of real-time applications.

The strength of the proposed scheme can be analysed, given the prime number y, an attacker can perform an after transmission attack using brute-force technique for searching the encryption key. Implementing the proposed scheme will make the attacker spend the same time for decrypting each packet, which the attacker used to spend for cracking the entire session packets in the existing methodologies. The attacker will have a hard time understanding and cracking the proposed encryption scheme.

Formula for calculating Chi- Square test:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O-E)^2}{E}$$

where O = Observed Frequency, E = Expected Frequency.

In the above table $P_{M,\ i}$ represents the packet number, where i ranges from 1 to n, n is the total no. of. Packets in a session. y is a prime number and α is a primitive root of prime number y. Both M and S share values of parameters y, α using D-H. X_M is the secret value of M and X_S is the secret value of S. Z_M is the blind key of M and Z_S is the blind key of S. K_M is the packet encryption key generated by M and K_S is the packet decryption key generated by S for each packet respectively using the other values in the table.

The efficiency of the proposed scheme depends on, how efficiently this scheme associates and balances both the attributes latency and security without compromising on either of them. For proving this association a model has been formulated and has been tested using chi-squared χ^2 test, hypothesis testing. Different keys have been used for each packet, by using the chi-squared test and based on the hypothesis testing it leads to the result that, there exists a good association between the set of attributes such as latency and security. Therefore, Null Hypothesis is accepted as per the Table 2 and the alternate hypothesis is rejected.

The encryption techniques used by existing methodologies for data security are proven to be secure as of now, but except WPA all other methods use the same key for encrypting all the data packets in a session. If a third party can decrypt a single packet during or after transmission, all the packets in that session can be compromised. In WPA each packet is encrypted using a different key but keys are generated by a combination that includes a base key, transmitting station MAC address, and the packet serial number, which is still vulnerable. These methodologies might be breakable in near future with the advancements happening in cloud computing. It is a proven fact that every encryption technique is breakable, but its efficiency depends on how much time it can with stand an attack. As in today's scenario, where cloud computing comes with almost unlimited computing resources, the time taken to crack an encryption technique with a single session key will gradually reduce and there might come a time when the present day sophisticated techniques can be compromised in seconds. The proposed scheme will make the attacker spend the amount same time for decrypting each packet, which the attacker used to spend for cracking the entire session in the existing methodologies.

4 Conclusion and Future work

This work proposes a packet encryption technique for MCC applications. The current encryption techniques, encrypt the entire packets in a session using the same session key. The exsisting encryption techniques may be efficient in providing data security for now, but as cloud technology advances these encryption techniques may not be sufficient in near future. In the proposed method, a packet key once used will never be used again in the same session balancing both security and latency. Packet encryption technique works efficiently against after transmission attacks and brute force attacks.

The strength of an encryption scheme depends on the key rather than the algorithm used for encryption, as the encryption algorithms become obsolete after a while, the packet encryption scheme has been proposed in such a way that it can be used with any encryption algorithm. The future work will formulate a detailed comparative study of the latency, security values of this proposed scheme with a particular encryption algorithm for different types of data. This will facilitate users to choose the encryption algorithm to be used with this scheme for their application.

References

1. Khan A u R, Othman M, Madani SA, Khan SU (2014) A Survey of Mobile Cloud Computing Application Models. IEEE Commun Surve Tutorials 16(1):393–413
2. Skourletopoulos G et al (2017) Towards Mobile Cloud Computing in 5G Mobile Networks: Applications, Big Data Services and Future Opportunities. In: Mavromoustakis C, Mastorakis G, Dobre C (eds) Advances in Mobile Cloud Computing and Big Data in the 5G Era. Studies in Big Data, vol 22. Springer, Cham
3. What is mobile cloud computing? (2013) https://www.ibm.com/blogs/cloud-computing/2013/06/mobile-cloud-computing/. Accessed: 14-10-2017
4. Mollah MB, Azad MAK, Vasilakos A (2017) Security and privacy challenges in mobile cloud computing: Survey and way ahead. J Netw Comput Appl 84:38–54
5. Akherfi K, Gerndt M, Harroud H (2018) Mobile cloud computing for computation offloading: Issues and challenges. Appl Comput Inf 14(1):1–16
6. Jung Y, Festijo E (2014) One-time packet key exchange scheme for secure real-time multimedia applications. J Comput Syst Sci 80: 1584–1596
7. Jung Y, Festijo E (2013) Securing RTP packets using per-packet selective encryption scheme for real-time multimedia applications: 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, Pages: 659–666
8. Jung Y, Festijo E, Foster JWA (2013) Securing RTP Packets Using Per-Packet Key Exchange for Real-Time Multimedia. ETRI J 35(Number 4):726–729
9. Rescorla E (1999) Diffie–Hellman Key Agreement Method, RFC2631 (proposed standard)
10. Freier et al., (2011) The Secure Sockets Layer (SSL) Protocol Version 3.0, RFC 6101
11. Dierks & Rescorla (2008) The Transport Layer Security (TLS) Protocol Version 1.2, RFC 5246
12. Zou Y, Zhu J, Wang X, Hanzo L (2016) A Survey on Wireless Security: Technical Challenges, Recent Advances, and Future Trends. Proc IEEE 104(9):1727–1765
13. Khasawneh M, Kajman I, Alkhudaidy R, Althubyani A (2014) A Survey on Wi-Fi Protocols: WPA and WPA2. In: Martínez Pérez G., Thampi S.M., Ko R., Shu L. (eds) Recent Trends in Computer Networks and Distributed Systems Security. SNDS 2014. Communications in Computer and Information Science, vol 420. Springer, Berlin, Heidelberg
14. Sheldon FT, Oak Ridge National Laboratory, Weber JM, Yoo S-M, Pan WD, (2012) The Insecurity of Wireless Networks, IEEE Security & Privacy Magazine, volume: 10, Issue: 4, Pages: 54–61
15. Hamzeh, et al., (1999) Point-to-Point Tunneling Protocol (PPTP), RFC 2637
16. Microsoft Windows CE 3.0 Support of PPP and PPTP (2000). Microsoft Corporation https://msdn.microsoft.com/en-us/library/ms834454.aspx. Accessed: 17-05-2018
17. Simion D, Ursuleanu F, Mihai G, Adrian P, Dan A, Lavric A (2012) Efficiency Consideration for Data Packets Encryption within Wireless VPN Tunneling for Video Streaming. Int J Comput Commun Control 8:136
18. Townsley, et al., (1999) Layer Two Tunneling Protocol "L2TP", RFC 2661
19. L2TP Tunnel Setup and Teardown (2018) https://www.cisco.com/c/en/us/support/docs/dial-access/virtual-private-dialup-network-vpdn/23980-l2tp-23980.html#t2. Accessed: 17-05-2018
20. Frankel K (2011) IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap, RFC 6071
21. Rosu SM, Popescu MM, Dragoi G, Guica IR (2012) The Virtual Enterprise Network based on IPSec VPN Solutions and Management, International Journal of Advanced Computer Science and Applications, Vol. 3, No. 11
22. VPNs and VPN Technologies (2002) http://www.ciscopress.com/articles/article.asp?p=24833&seqNum=3. Accessed: 17-05-2018

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.