QOS based Wireless Body Area Network (WBAN) for Healthcare

P. Kavitha[1], R. Jayasanthi[2], S. Meenakshi Ishwaryar[3]
[1]*Assistant Professor, Dept. of CSE, P. S. R. Rengasamy College of Engineering for Women, Sivakasi, India*
[2,3]*UG Student, Dept. of CSE, P. S. R. Rengasamy College of Engineering for Women, Sivakasi, India*

***Abstract*: In many military network scenarios, connections of wire- less devices carried by soldiers may be temporarily disconnected by jamming, environmental factors, and mobility, especially when they operate in hostile environments. In a hospital environment, the total number of Wireless Body Area Network (WBAN) equipped patients requesting ubiquitous healthcare services in an area increases significantly. Therefore, increased traffic load and group-based mobility of WBANs degrades the performance of each WBAN significantly, concerning service delay and network throughput. In addition, the mobility of WBANs affects connectivity between a WBAN and an Access Point (AP) dynamically, which affects the variation in link quality significantly. To address the connectivity problem and provide Quality of Services (QoS) in the network, we propose a dynamic connectivity establishment and cooperative scheduling scheme, which minimizes the packet delivery delay and maximizes the network throughput. First, to secure the reliable connectivity among WBANs and APs dynamically, we formulate a selection parameter using a price-based approach.**

***Keywords*: Wireless Body Area Network, Biomedical monitoring, Smart Health, QoS, Cooperative Packet Scheduling, Dynamic Connectivity Establishment, Coalition Game Theory, Performance Analysis.**

1. Introduction

Delay and Disruption Tolerant Networks (DTNs) are networks that aim to bring low-cost best-effort connectivity to challenged environments with no or limited infrastructures. Nodes in DTNs are often highly mobile and experience intermittent connectivity. DTNs can be deployed in developing countries and are poised to play a key part in future space networks.

A. *No end to end path*

- Node mobility creates partitions in the network. We cannot assume that there is a complete end to end path between a source and destination.
- If a path does exist it is assumed to be unstable. Instead, an end to end path exists over time, as nodes move and forward messages to each other.

B. *High message delays*

- The opportunistic nature of DTNs means messages that are delivered often experience high delays.
- Delays can be typically on the order of minutes or hours, but could potentially be days depending on the exact scenario.

C. *Objectives*

Asynchronously interconnecting different networks

- Network of regional networks Network of regional networks
- Each network can have
- Arbitrary underlying technologies
- Different administrative controls
- No accessible infrastructure

2. Existing system

- In Existing System, they used state of the art algorithm for creating network.
- In a link-failure situation, it is very crucial to provide dynamic connectivity to critical WBANs among multiple coexisting WBANs to send the former's medical data with minimum packet delivery delay.
- Interference from coexisting WBANs degrades the performance of each WBAN in terms of networks throughput and energy consumption is major problem.
- Existing pieces of literature assume that WBANs connect to APs contentiously, but due to change in link states, connectivity gets affected.

A. *Disadvantages*

- In existing literature noticed the problem of cooperative communication in the theme of WBANs.
- WBANs send data to medical experts through the existing communication infrastructures such as cellular networks, Wi-Fi, and IEEE 802.15.6-based networks.

B. *Proposed system*

- In proposed a Clique-Based WBAN Scheduling (CBWS) algorithm, in which WBANs are partitioned into different groups, which are activated in different time slots.
- In this work we improve the existing state-of-the-art by proposing a solution for ensuring continuous

connectivity in WBANs even in presence of group mobility of nodes and body shadowing effects during medical emergency situations.

- The existing models only consider the homogeneous traffic flows for the data communication process, but the proposed model is suitable to heterogeneous traffic flows in the network.
- The existing models do not consider the group-based mobility, whereas the proposed model considers group-based mobility in order to incorporate the effects of dynamic postural partitioning and variation of link-qualities in the network.

C. *Advantages*

- The proposed coalition approach, the average payoff of each WBAN increases significantly and the critical WBANs get less packet delivery delay.
- In the proposed scheme, the critical WBANs transmit their packets to the AP using cooperative scheduling.
- Therefore, the critical WBANs consume less energy than the normal ones. With the increase in the number of APs, the energy consumption of each WBAN decreases.

D. *List of modules*

- Network Architecture
- CP-ABE Scheme Construction
- TBSSM Scheme Construction
- Revocation
- Key Updation

1) *Network architecture*

- Since the key authorities are semi-trusted, they should be deterred from accessing plaintext of the data in the storage node; meanwhile, they should be still able to issue secret keys to users.
- In order to realize this somewhat contradictory requirement, the central authority and the local authorities engage in the arithmetic 2PC protocol with master secret keys of their own and issue independent key components to users during the key issuing phase.
- The 2PC protocol prevents them from knowing each other's master secrets so that none of them can generate the whole set of secret keys of users individually.
- Thus, we take an assumption that the central authority does not collude with the local authorities (otherwise, they can guess the secret keys of every user by sharing their master secrets).

2) *CP-ABE scheme construction*

- In CP-ABE, user secret key components consist of a single personalized key and multiple attribute keys.
- The personalized key is uniquely determined for each user to prevent collusion attack among users with different attributes.
- The proposed key generation protocol is composed of the personal key generation followed by the attribute key generation protocols.
- It exploits arithmetic secure 2PC protocol to eliminate the key escrow problem such that none of the authorities can determine the whole key components of users individually.
- Personal Key Generation: The central authority and each local authority is involved in the following protocol. When a sender wants to deliver its confidential data, he defines the tree access structure over the universe of attributes, encrypts the data under to enforce attribute-based access control on the data, and stores it into the storage node.
- Then a user receives the ciphertext from the storage node, the user decrypts the ciphertext with its secret key.
- The algorithm performs in a recursive way.
- We first define a recursive algorithm that takes as inputs a cipher text, a private key, which is associated with a set of attributes, and a node from the tree.

3) *TBSSM scheme construction*

- Unauthorized access of data, cloud made unreliable for client.
- To provide reliability on cloud, an approach TBSSM is advised at client end to make safe and secure storage of data.
- The proposed approach is stack of multiple protections layer that deals with clients' data to providing overlapping layers of authentication, behavior analysis and make data unreadable form using behavior based encryption mechanisms.
- The suggested TBSSM approach consists of several phases. Firstly, to coated authentication layer to the data by providing identity of users and verifying the claimed identity.
- Secondly to coated behavior analysis layer to the data by regular observing the activities of users on the basis of historical property.
- Third phase is to coated behavior based encryption layer by converting client data into encrypted data and send for storage on DTN.

4) *Revocation*

- We observed that it is impossible to revoke specific attribute keys of a user without rekeying the whole set of key components of the user in ABE key structure since the whole key set of a user is bound with the same random value in order to prevent any collusion attack.
- Therefore, revoking a single attribute in the system requires all users who share the attribute to update all their key components even if the other attributes of them are still valid.

- This seems very inefficient and may cause severe overhead in terms of the computation and communication cost, especially in large-scaled DTNs.

5) Key update

- When a user comes to hold or drop an attribute, the corresponding key should be updated to prevent the user from accessing the previous or subsequent encrypted data for backward or forward secrecy, respectively.
- The key update procedure is launched by sending a join or leave request for some attribute group from a user who wants to hold or drop the attribute to the corresponding authority.
- On receipt of the membership change request for some attribute groups, it notifies the storage node of the event. Without loss of generality, suppose there is any membership change in (e.g., a user comes to hold or drop an attribute at some time instance).

3. System specification

A. Hardware specification

This section describes the hardware components with which the tool was developed and the minimum hardware configuration with which the system operates best.

Processor	: Intel Core 2 Duo
RAM	: 2GB
Mother Board	: Intel Board
Hard Disk	: 80 GB Hard Disk

B. Software specification

This section describes the software in which the application was developed and using the same software would make it more compatible.

Operating System	: Windows 7
Language	: Java
Back End	: MySQL

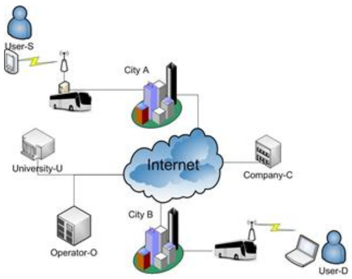

Fig. 1. Internet

C. Applications

1) Verifying the integrity of files or messages

An important application of secure hashes is verification of message integrity. Determining whether any changes have been made to a message (or a file), for example, can be accomplished by comparing message digests calculated before, and after, transmission (or any other event). For this reason, most digital signature algorithms only confirm the authenticity of a hashed digest of the message to be "signed". Verifying the authenticity of a hashed digest of the message is considered proof that the message itself is authentic. MD5, SHA1, or SHA2 hashes are sometimes posted along with files on websites or forums to allow verification of integrity [3]. This practice establishes a chain of trust so long as the hashes are posted on a site authenticated by HTTPS.

2) Password verification

A related application is password verification (first invented by Roger Needham). Storing all user passwords as cleartext can result in a massive security breach if the password file is compromised. One way to reduce this danger is to only store the hash digest of each password. To authenticate a user, the password presented by the user is hashed and compared with the stored hash. (Note that this approach prevents the original passwords from being retrieved if forgotten or lost, and they have to be replaced with new ones.) The password is often concatenated with a random, non-secret salt value before the hash function is applied. The salt is stored with the password hash. Because users have different salts, it is not feasible to store tables of precomputed hash values for common passwords. Key stretching functions, such as PBKDF2, Bcrypt or Scrypt, typically use repeated invocations of a cryptographic hash to increase the time required to perform brute force attacks on stored password digests.

3) File or data identifier

A message digest can also serve as a means of reliably identifying a file; several source code management systems, including Git, Mercurial and Monotone, use the sha1sum of various types of content (file content, directory trees, ancestry information, etc.) to uniquely identify them. Hashes are used to identify files on peer-to-peer filesharing networks. For example, in an ed2k link, an MD4-variant hash is combined with the file size, providing sufficient information for locating file sources, downloading the file and verifying its contents. Magnet links are another example. Such file hashes are often the top hash of a hash list or a hash tree which allows for additional benefits. One of the main applications of a hash function is to allow the fast look-up of a data in a hash table. Being hash functions of a particular kind, cryptographic hash functions lend themselves well to this application too.

However, compared with standard hash functions, cryptographic hash functions tend to be much more expensive computationally. For this reason, they tend to be used in contexts where it is necessary for users to protect themselves against the possibility of forgery (the creation of data with the same digest as the expected data) by potentially malicious participants. Pseudorandom generation and key derivation. Hash functions can also be used in the generation of pseudorandom bits, or to derive new keys or passwords from a single, secure key or password.

Hash functions based on block ciphers: There are several methods to use a block cipher to build a cryptographic hash function, specifically a one-way compression function. The methods resemble the block cipher modes of operation usually used for encryption. All well-known hash functions, including MD4, MD5, SHA-1 and SHA-2 are built from block-cipher-like components designed for the purpose, with feedback to ensure that the resulting function is not invertible. SHA-3 finalists included functions with block-cipher-like components (e.g., Skein, BLAKE) though the function finally selected, Keccak, was built on a cryptographic sponge instead.

A standard block cipher such as AES can be used in place of these custom block ciphers; that might be useful when an embedded system needs to implement both encryption and hashing with minimal code size or hardware area. However, that approach can have costs in efficiency and security. The ciphers in hash functions are built for hashing: they use large keys and blocks, can efficiently change keys every block, and have been designed and vetted for resistance to related-key attacks. General-purpose ciphers tend to have different design goals. In particular, AES has key and block sizes that make it nontrivial to use to generate long hash values; AES encryption becomes less efficient when the key changes each block; and related-key attacks make it potentially less secure for use in a hash function than for encryption.

4) Merkle–Damgård construction

A hash function must be able to process an arbitrary-length message into a fixed-length output. This can be achieved by breaking the input up into a series of equal-sized blocks, and operating on them in sequence using a one-way compression function. The compression function can either be specially designed for hashing or be built from a block cipher. A hash function built with the Merkle–Damgård construction is as resistant to collisions as is its compression function; any collision for the full hash function can be traced back to a collision in the compression function.

The last block processed should also be unambiguously length padded; this is crucial to the security of this construction. This construction is called the Merkle–Damgård construction. Most widely used hash functions, including SHA-1 and MD5, take this form.

The construction has certain inherent flaws, including length-extension and generate-and-paste attacks, and cannot be parallelized. As a result, many entrants in the current NIST hash function competition are built on different, sometimes novel, constructions.

5) Use in building other cryptographic primitives

Hash functions can be used to build other cryptographic primitives. For these other primitives to be cryptographically secure, care must be taken to build them correctly. Message authentication codes (MACs) (also called keyed hash functions) are often built from hash functions. HMAC is such a MAC. Just as block ciphers can be used to build hash functions, hash functions can be used to build block ciphers. Luby-Rackoff constructions using hash functions can be provably secure if the underlying hash function is secure. Also, many hash functions (including SHA-1 and SHA-2) are built by using a special-purpose block cipher in a Davies-Meyer or other construction. That cipher can also be used in a conventional mode of operation, without the same security guarantees. See SHACAL, BEAR and LION.

Pseudorandom number generators (PRNGs) can be built using hash functions. This is done by combining a (secret) random seed with a counter and hashing it. Some hash functions, such as Skein, Keccak, and RadioGatún output an arbitrarily long stream and can be used as a stream cipher, and stream ciphers can also be built from fixed-length digest hash functions. Often this is done by first building a cryptographically secure pseudorandom number generator and then using its stream of random bytes as keystream. SEAL is a stream cipher that uses SHA-1 to generate internal tables, which are then used in a keystream generator more or less unrelated to the hash algorithm. SEAL is not guaranteed to be as strong (or weak) as SHA-1. Similarly, the key expansion of the HC-128 and HC-256 stream ciphers makes heavy use of the SHA256 hash function.

4. Conclusion

This paper presented an overview of QOS based Wireless Body Area Network (WBAN) for health care.

References

[1] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, “Maxprop: Routing for vehicle-based disruption tolerant networks,” in Proc. IEEE INFOCOM, 2006, pp. 1–11.

[2] M. Chuah and P. Yang, “Node density-based adaptive routing scheme for disruption tolerant networks,” in Proc. IEEE MILCOM, 2006, pp. 1–6.

[3] M. M. B. Tariq, M. Ammar, and E.Zequra, “Message ferry route de- sign for sparse ad hoc networks with mobile nodes,” in Proc. ACM MobiHoc, 2006, pp. 37–48.

[4] S. Roy and M. Chuah, “Secure data retrieval based on cipher text policy attribute-based encryption (CP-ABE) system for the DTNs,” Lehigh’s Tech. Rep., 2009.

[5] M. Chuah and P. Yang, “Performance evaluation of content-based information retrieval schemes for DTNs,” in Proc. IEEE MILCOM, 2007, pp. 1–7.

[6] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, “Plutus: Scalable secure file sharing on untrusted storage,” in Proc. Conf. File Storage Technol., 2003, pp. 29–42.

[7] L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, “Mediated ciphertext-policy attribute-based encryption and its application,” n Proc. WISA, 2009, LNCS 5932, pp. 309–323.

[8] N. Chen, M. Gerla, D. Huang, and X. Hong, “Secure, selective group broadcast in vehicular networks using dynamic attribute based encryption,” in Proc. Ad Hoc Netw. Workshop, 2010, pp. 1–8.

[9] D. Huangand M. Verma, “ASPE: Attribute-based secure policy enforcement in vehicular ad hoc networks,” Ad Hoc Netw., vol. 7, no. 8, pp. 1526–1535, 2009.

[10] A. Lewko and B. Waters, “Decentralizing attribute-based encryption,” Cryptology ePrint Archive: Rep. 2010/351, 2010.