Secure End-to-End VoLTE based on Ethereum Blockchain

Elie F. Kfoury and David J. Khoury
Department of Computer Science, American University of Science and Technology, Beirut, Lebanon
Email: {ekfoury, dkhoury}@aust.edu.lb

***Abstract*—Voice over Long Term Evolution (VoLTE) technology defines standards to deliver real-time services such as voice and video over LTE based on IP Multimedia Subsystem (IMS) networks. The security implementation in VoLTE is End-to-Access (e2a), which means that the sessions are only encrypted between the mobile terminals and the IMS network. In this paper we propose a new approach for securing End-to-End (e2e) VoLTE media based the Ethereum Blockchain. The solution consists of creating public and private keypairs for VoLTE user equipments (UEs) and storing the public keys in the Ethereum Blockchain. The media is encrypted e2e using the Secure Real Time Protocol (SRTP) protocol with a variety of session key distribution mechanisms. Results showed that the solution implementation has minimal impact on the existing IMS network, and the secure call setup time between two terminals is negligible compared to the original VoLTE setup time.**

***Keywords*—Blockchain; Ethereum; IMS; LTE; SRTP; VoLTE.**

I. Introduction

Voice over Long-Term Evolution (VoLTE) is a cutting-edge technology that provides real-time services over the LTE mobile network. This technology relies on the IP Multimedia Subsystem (IMS) [1] platform which uses the Session Initiation Protocol (SIP) [2] as its core protocol. Applications that are likely to be offered by IMS networks include Voice telephony, Video calls, Instant Messaging (IM), conference calls, Push-to-Talk (PTT), and many others. Ensuring security and privacy in such applications is of utmost importance as eavesdroppers and malicious parties are able to intercept the multimedia session and violate user's privacy. VoLTE sessions by default are not end-to-end encrypted [3], hence, it is easy to eavesdrop on a connection, especially by mobile network operators (MNO).

Blockchain on the other hand is an emergent technology that provides decentralization with no single point of failure and ensures data immutability through cryptographic functions and consensus algorithms and protocols. The Ethereum Blockchain is an open-source distributed computing platform featuring smart contract (scripting) functionality. Developers can easily write decentralized applications on high level and benefit from the distribution inherited from the Blockchain technology.

In this paper, we propose a novel approach for securing VoLTE media peer-to-peer based on the Ethereum Blockchain by providing a trustless key distribution management method. The main contributions of this paper include: 1) Implementing End-to-End security for a variety of VoLTE applications, 2) Providing transparency when interacting with the Blockchain, 3) Introducing new business models for mobile network operators.

The paper starts with an introduction on Blockchain technology and VoLTE systems. Section II describes VoLTE and its security measures, Section III presents our proposed system, Section IV discusses the results of the simulation, Section V provides discussions and Section VI concludes with the intended future work.

II. Background on VoLTE and Security

VoLTE is a Voice over IP (VoIP) service specified in the GSMA Permanent Reference Document (PRD) IR.92 [4]. LTE provides VoLTE and high-speed data services simultaneously. Unlike Internet-based VoIP, which works on a "best effort" basis, VoLTE uses the IMS platform and new radio access features to provide low latency, better quality of service (QoS), and others. The IMS platform is based on SIP protocol which is used for session initiation and control. Interoperability in VoLTE with legacy networks, namely, the circuit switched networks, is supported through MGWs. Fig. 1 shows an overview architecture of VoLTE. This technology is expected to replace the current circuit switched telephony gradually with the increase coverage deployment of LTE. It was first deployed in Asia (South Korea and Singapore), and later by US MNOs (Verizon, ATT). The GSA (Global mobile Suppliers Association) published in early 2017 the worlds map illustrating the number of MNOs deploying VoLTE [5].

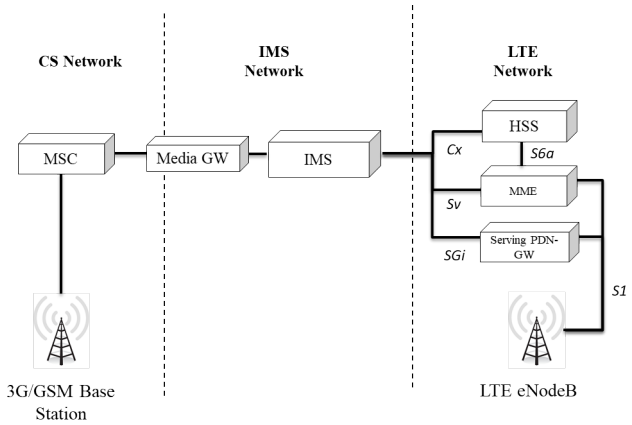

Fig. 1. VoLTE Overview Architecture

A. IMS Architecture

In this section we introduce the logical components of an IMS network, and explain the role of each component (Fig. 2) when setting up a VoLTE session.

1) The P-CSCF: The Proxy Call Session Control Function (P-CSCF) is considered as the user's entry point to the IMS network. The signaling messages sent from the user (Registration and call setup) should pass through the P-CSCF. This component is responsible of SIP signaling compression, Quality of Service (QoS) control, Billing, and end-to-access security through IP Security (IPsec).

2) The S-CSCF: The Serving Call Session Control Function (S-CSCF) is the SIP Server (SIP Registrar, SIP Proxy) in an IMS network. It provides session management functions and is in charge of routing the SIP messages between the endpoints.

3) The I-CSCF: The Interrogating Call Session Control Function (I-CSCF) is used to route the first SIP request to the right S-CSCF. It is done by interrogating the Home Subscriber Server (HSS).

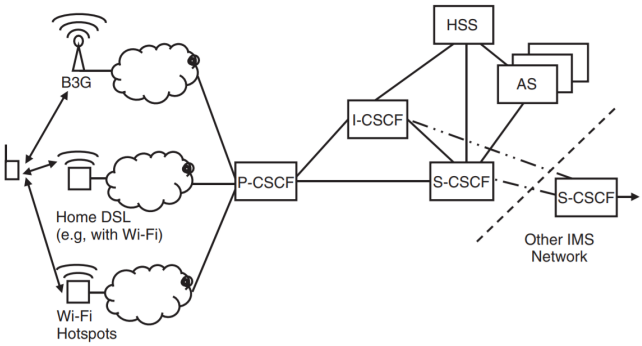

Fig. 2. IMS Architecture src: "3G, 4G and Beyond", Martin Sauter

4) The HSS: The Home Subscriber Server (HSS) is the database that contains the users' IMS profiles and other information pertaining to the subscriber. It also includes the security parameters of the subscribers which are located in the user's Subscriber Identity Module (SIM) card.

B. VoLTE Security

The LTE uses the IMS network to deliver VoLTE services, based on SIP messages. This makes the IMS network act as a SIP proxy, performing routing, session control and registration. Media or Voice are delivered through the Real Time Protocol (RTP) [6] from one User Equipment (UE) to the other. The security implementation in VoLTE is End-to-Access (e2a), which means that the payload is only ciphered between the mobile terminal and the P-CSCF and not End-to-End (e2e) between the two terminals. The control signaling is secured by establishing IPsec, a secure tunnel between the UE and IMS through PCSCF. As long as the user is connected to the P-CSCF, this secure tunnel remains established. In VoLTE, IPSec uses the transport mode to encapsulate IP payload (Encryption Integrity Protection Security of Signaling Traffic defined in 3GPP TS 133.203). Media Protection is specified in 3GPP TS 133.328, but optionally provided as e2a by using SRTP protocol. In the next section we introduce our proposed system for secure e2e VoLTE calls based on Ethereum Blockchain.

III. Proposed System

The main problem with VoLTE is the lack of end-to-end security. In our solution, we intend to provide this feature by relying on the Ethereum Blockchain network to store and manage the VoLTE subscribers' public keys. Fig. 3 illustrates the overview architecture and the interfaces in the system.

A. System Architecture and Components

We describe briefly on high level the system architecture, interfaces and the impact on the exiting VoLTE implementation. The proposed solution uses the Ethereum Blockchain to store the public keys of the VoLTE components in the Ethereums deployed smart contract. Hence, the Ethereum Blockchain acts as a trusted distributed public key store. The IMS platform is impacted by adding two new modules: Ethereum IMS Wallet Management and an Ethereum Full Node to interface the Blockchain. We refer to this modified platform as Ethereum IMS (EIMS). The main roles of the EIMS wallet management are: Transferring Ether to the VoLTE client and approving its public key storage transaction. Two new modules are added to the existing VoLTE client: the Light Ethereum Subprotocol (LES) [7] and the wallet management interface to the EIMS. The high-level sequence of events is as follows: The IMS stores its public key Pu_{EIMS} in the Smart Contract deployed on the Blockchain network (1). The VoLTE client gets a new secure SIM card that contains a token signed by the private key of the operators EIMS (2). The purpose of the token is to ensure the subscriber authenticity in requiring Ether from EIMS. Once the SIM card is inserted into the UE, an empty Ethereum wallet is generated. The VoLTE client fetches Pu_{EIMS} from the Smart Contract (3), and requests Ether from the EIMS (4). After validation of the request and the successful transfer of Ether by the EIMS (5), the VoLTE client stores its generated keypair (Pu_A and Pr_A) on the SIM card and Pu_A in the Blockchain (6).

1) Secure VoLTE Client: The Secure VoLTE client comprises the following components: a standard VoLTE client, a light Ethereum client that connects to the Blockchain through LES, and a dedicated module (Eth Wallet) to interwork with the EIMS function. At the initial setup of the secure SIM, an empty Ethereum wallet is generated by the VoLTE client. Transferring Ether to the client's wallet is mandatory as storing data or changing the state of a contract in the Blockchain requires transaction fees priced in Ether. Therefore, the VoLTE client requests Ether from the EIMS by sending its newly generated wallet address, Mobile Station International Subscriber Directory Number (MSISDN), and the signed token as an encrypted payload (interface 4 in Fig. 3). This interface is encrypted using TLS-PSK (Pre-Shared Keys). The distribution of the pre shared key between the two entities is based on

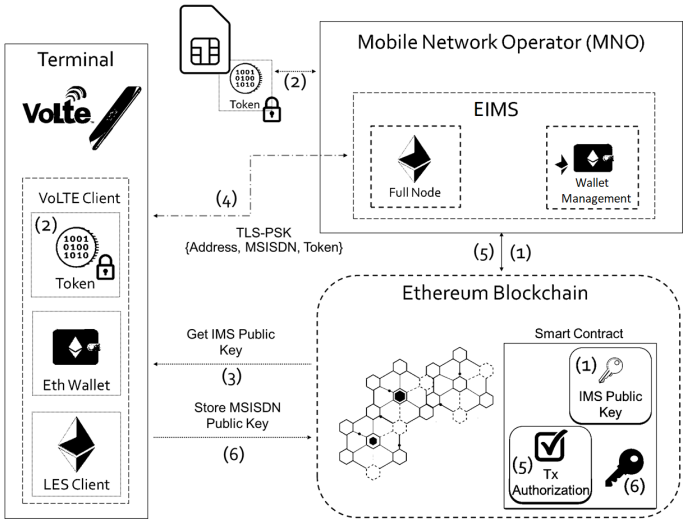

Fig. 3. Proposed System Architecture

the Needham-Schroeder method [8] as illustrated in Fig. 4. The EIMS validates the request by verifying its signature on the token and transfers Ether to the client through the Blockchain network (interface 5). Upon successful transfer, the client stores its public key, the token value, and the MSSIDN in the contract (interface 6).

2) Smart Contract: Smart contract is an account holding object that contains distributed code executed by the Blockchain network. Solidity [9] is the contract-oriented high-level programming language used to write smart contracts that can be used with the Ethereum Virtual Machine (EVM)[10][11]. A mapping data structure indexed by the token is used in our contract to map between MSISDN and the subscriber's public key. The smart contract functions include: a) *addClient(MSISDN, Pu_{sub}, Token):* This function call enables any user to insert a record into the Blockchain containing its MSISDN, public key, and Token. However, identity theft vulnerability exist here: an attacker might register a record of a legitimate user and impersonate him. To overcome this weakness, two mappings are created: pending_list and approved_list. The pending list contains record of any user calling the addClient function, and the approved list contains records that are moved from the pending list by the EIMS wallet management. b) *approveClient(MSISDN, Token)* This function moves the subscriber's entry from the pending list to the approved list. It is programmed such that only the EIMS wallet management is able to execute it. As a results, the risk of approving clients by unauthorized users is eliminated. c) *getClient(MSISDN)* This function call accepts as parameter the MSISDN of the destination and returns the corresponding public key from the approved list.

3) EIMS Wallet Management: The fundamental functionalities of the EIMS wallet management are: *a) Ether Transfer:* Upon receiving the encrypted payload mentioned in the Secure VoLTE client section, the EIMS verifies its signature applied on the token. The aim of this verification is to ensure that the subscriber requesting Ether has purchased the secure VoLTE SIM. On successful verification, it sends sufficient ether to the wallet address received through the encrypted payload. *b) Storage Transaction Approval:* After invoking the addClient function by the VoLTE client, it requests approval from the EIMS wallet management. This approval is essential as there is no control on the storage in the Blockchain network. Moreover, a malicious subscriber cannot overwrite a previously stored record even if the stored token is compromised. Finally, since the EIMS wallet management is the contract deployer, it can approve the subscriber by moving its entry from the pending list to the approved list.

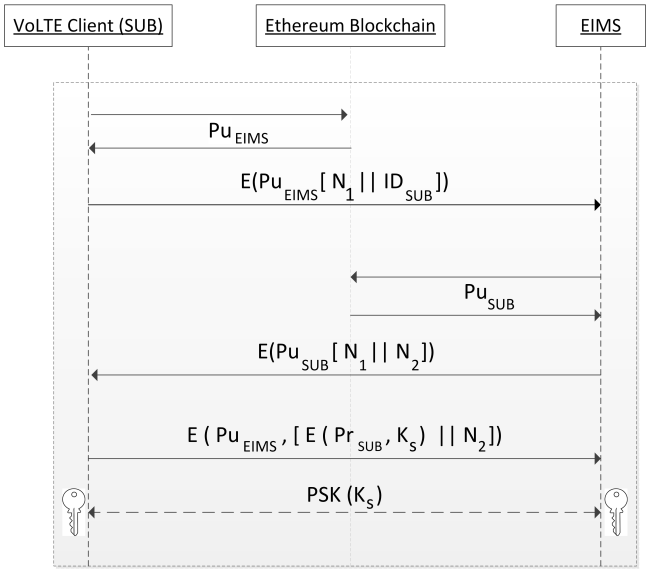

Fig. 4. Pre-Shared Key distribution using Needham-Schoreder Method

B. Ethereum Light Client Security

Knowing that this solution will be primarily used on mobile devices, downloading the whole Ethereum Blockchain (∼390 GB Dec 2017) is not feasible. Consequently, the light client is developed to enable Ethereum nodes to run on all types of machines (smart phones, Internet of Things (IoT) devices, laptops ...). The chaindata of the light client is approximately 0.005GB, which is feasible on mobile phones especially that this chaindata is downloaded once for all types of Ethereum decentralized applications. Merkle trees play an important role in ensuring light client security. They represent a data structure that allow efficient and secure verification of data. The client requests a light client server, which in turn, fetches the Merkle branch and sends it back to the client. More on Merkle Trees is found in [12]. The asymptotic complexity of data retrieval from Merkle trees is logarithmic.

C. End to End Secure VoLTE System

The main changes imposed by this system are discussed in the subsection A. The security solution uses the Ethereum Blockchain to hold the IMS and subscribers' public keys.

These keys will be used to produce session keys that secure the media between two VoLTE clients. This subsection is divided into two main parts:

1) SIP Signaling Security: The SIP signaling between the UE and the P-CSCF is secured e2a using IPSec. After establishing this encrypted tunnel, the VoLTE client sends a SIP REGISTER message to the SIP server to register itself if authorized. All remaining SIP messages for establishing a call between two endpoints are e2a encrypted.

2) End to End Media Security: The session establishment between two VoLTE UEs is done according to 3GPP IMS standard without any changes as described previously, which results in payload session establishment between the two UEs. The end to end secure call can be set by generating a random session key to encrypt the RTP payload between the two UEs using the SRTP protocol. One of the solutions to generate and distribute the session keys is to use the Station-to Station (STS) [13] key agreement scheme as demonstrated in Fig. 5. The main advantage of this approach is the usage of Diffie Hellman (DH) as a key exchange (Perfect Forward secrecy). The issue of public keys distribution in DH is solved by storing them into the Blockchain. The VoLTE UE A caller client fetches the public of the VoLTE UE B callee from the Ethereum and vice versa. The UE A and UE B exchanges the DH parameters through RTP Control Protocol (RTCP) protocol [6] as shown in Fig. 5.

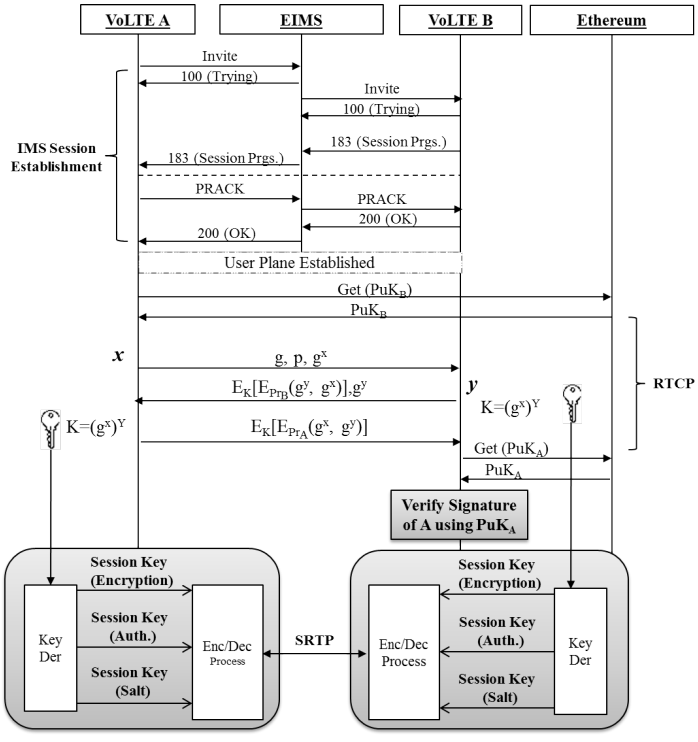

Fig. 5. Secure VoLTE call setup using STS method

IV. Results and Simulation

In a preliminary study [14], we simulated the solution using VoIP technology instead of VoLTE. As VoLTE is not yet deployed in Lebanon, we rely on the results of the previous study and estimate the changes on VoLTE.
The Ethereum Testnet is the used Blockchain implementation. Storing the subscribers' public keys took 11.5 seconds to be validated. This amount of time is spent only once per subscriber upon registration. The call setup took 1218 ms to secure the call between two VoIP subscribers. This includes key retrieval from the Blockchain, Diffie Hellman key exchange, SRTP key derivation. P3 Communications found that in VoLTE, call set-up time ranges from 2 to 4 seconds [15]. Table I shows the results of VoIP and the expected VoLTE call setup times.

TABLE I. Call Setup Time (VoIP and VoLTE)

VoIP Call Setup Time	Expected VoLTE Call Setup Time
1218 ms	3218 ms - 5218 ms

V. Discussions

It is worth noting that subscribers from different mobile operators can establish secure e2e VoLTE calls with the condition that the IMS of every operator implements this proposed solution.
This system is currently limited in cases where the call is established between VoLTE and circuit switched (CS) networks. This concern is being investigated on an ongoing research project to enable end-to-end security and interoperability between these networks. According to Ericsson Mobility Report (November 2017), VoLTE technology will be the foundation for enabling voice calls over 5G access networks. We expect that VoLTE over 4G and 5G access will be controlled by the IMS platform as the IMS is access-agnostic.

VI. Conclusion

In this paper we have presented a novel approach to implement end to end media security for VoLTE subscribers without impacting the IMS standard of call set up. The impact on IMS platform consists of adding an Ethereum full node to interface the Ethereum Blockchain and a wallet management. A new secure SIM is needed for the VoLTE users and the client should be modified to include the LES and a wallet management. The secure e2e VoLTE presents a new service for the MNO which could be offered to VIP users requiring ultimate voice security. Regarding the future work, we intend to test our system on a deployed VoLTE network to consolidate our results. The next step is to standardize the proposed system and extend it to support different types of IMS applications.

REFERENCES

[1] IP Multimedia Subsystem. stage 2. 3gpp ts 23.228, 2006.

[2] Jonathan Rosenberg, Henning Schulzrinne, Gonzalo Camarillo, Alan Johnston, Jon Peterson, Robert Sparks, Mark Handley, and Eve Schooler. Sip: session initiation protocol. Technical report, 2002.

[3] IP Multimedia Subsystem. media plane security (3gpp ts 33.328 version 9.2. 0 release 9), jun. 18, 2010. *ETSI TS*, 133(328):V9.

[4] IR GSMA. Ims profile for voice and sms, 92.

[5] GSA. Volte service description and implementation guidelines, http://www.voiceage.com/pdfs/170131-SNAPSHOT-VoLTE_January_2017.pdf 2017.

[6] Van Jacobson, Ron Frederick, Steve Casner, and H Schulzrinne. Rtp: A transport protocol for real-time applications. 2003.

[7] Light client protocol ethereum/wiki wiki github.

[8] Gavin Lowe. Breaking and fixing the needham-schroeder public-key protocol using fdr. In *International Workshop on Tools and Algorithms for the Construction and Analysis of Systems*, pages 147–166. Springer, 1996.

[9] Chris Dannen. *Introducing Ethereum and Solidity*. Springer, 2017.

[10] Vitalik Buterin et al. Ethereum white paper. *GitHub repository*, 2013.

[11] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 151:1–32, 2014.

[12] Ralph C Merkle. Digital signature system and method based on a conventional encryption function, November 14 1989. US Patent 4,881,264.

[13] Simon Blake-Wilson and Alfred Menezes. Unknown key-share attacks on the station-to-station (sts) protocol. In *International Workshop on Public Key Cryptography*, pages 154–170. Springer, 1999.

[14] Elie Kfoury David Khoury. Secure end-to-end VoIP based on Ethereum Blockchain. *Journal of Communications - PREPRINT*.

[15] P3 CommsDay Mobile Benchmark. A special report analysing the 2016 data.