EasyChair Preprint

№ 10386

# Survey of Key Management and Secure Bootstrapping for Large Scale Constrained-Node Network

Sharib Rizwan and Sameer Soni



June 13, 2023

# Survey of Key Management and Secure Bootstrapping for Large Scale Constrained-Node Network

In the guidance of Prof. Dr. Oliver Hahm

*Frankfurt University of Applied Sciences*, Frankfurt am Main, Germany

Sharib Rizwan
Matr.-No.: 1386422
*dept. of High Integrity Systems*
sharib.rizwan@stud.fra-uas.de

Sameer Soni
Matr.-No.: 1392911
*dept. of High Integrity Systems*
soni.sameer@stud.fra-uas.de

***Abstract*—In today's world, sensors and devices have managed to bring the physical world into the digital world. They are everywhere, name any field be it healthcare, defense, lifestyle, safety or security. Sensors and connected devices are becoming part of life. As they influence so many activities and their impact is huge, it is very important that they are secure from an outer attack. Security is a very important and prime aspect of the IoT devices and network. Generally, these devices are very small and have limited computational, power and memory due to which it is not possible to use standard conventional security protocols with these devices for active security. In this paper, we will explore various techniques proposed for secure bootstrapping and key management schemes for this kind of constrained devices/nodes and networks.**

***Index Terms*—sensor, constrained node, constrained-node network, secure bootstrapping, key management schemes, network, internet of things**

## I. Introduction

Every day thousands of devices are getting connected to provide Internet of Things (IoT) services over the internet. This is expected to grow more and billions of IoT devices will be connected to IoT services in the near future. These devices connect the physical world to the digital world by sensing, observing, and identifying physical parameters and converting to digital attributes and transmitting over the network, generally over the internet to perform desired tasks. There are no fields left where IoT devices and applications are not getting utilized. Healthcare, defense, transportation, smart2 homes and smart buildings are some of the leading industries for IoT. This convergence of cyber and physical worlds connects people, things, and data create enormous opportunities.

IoT network is made up of heterogeneous devices and technologies posing varying capabilities. These devices can be large, powerful, or very small having multi-faceted constraints. Generally, IoT devices are very small and have limited ca-capabilities in terms of processing power, memory, and power resources. In other words, these devices are constrained in nature, so-called constrained nodes in network terms. Due to their limited capabilities, often lead to constraints on their own network, known as Constrained-Node network.

Security is most important in IoT technology to ensure secure communication of message/data, the authenticity of devices and integrity. There are various standard protocols and methods available like IPSec, TLS, and DTLS to achieve earlier mentioned goals but these are very expensive and resource mongering solutions which are not suitable for the constrained-node IoT networks. Along with these challenges, secure bootstrapping is also another challenge. Bootstrapping can be defined as the procedure by which an IoT device gets the secret keys and URL for reaching the necessary servers. This is a stage at which devices establish security associations including attributes like cryptographic algorithm and its mode, security keys and other required network parameters. Key management includes the generation, exchange, storage, usage and replacement of keys. These are two very important aspects of security in IoT devices. In this survey paper, we will be studying various methods and solutions to achieve secure bootstrapping and key management for the constrained-node network. [1] [2] [3]

### A. Paper Outline:

The outline of the article is as follows. In the section II we discuss the classification of various secure bootstrapping methods for a constrained-node network which is based on various approaches like key delivery mechanism, cryptographic method and authentication method. In section III we will be discussing the secure Key Bootstrapping Protocols mainly certificate-based and certificate-less approaches. Going forward, in section IV we will look into a different key management schemes for constrained-node network environments.

## II. Classification of Key Bootstrapping approaches for constrained nodes

Various Key bootstrapping protocols, some of which we will discuss further in our paper can be classified in different categories based on their underlying encryption algorithms,

key delivery mechanism and authentication mechanism. We will briefly discuss these in next sections.

### A. Key Delivery Scheme

In the context of Key Delivery, bootstrapping is further classified into two categories. (i) Key Transport - Key transport is based on the method of transportation of secret keys securely. These protocols can be symmetric, asymmetric, or other cryptographic algorithms (ii) Key Agreement - Key agreement components are those in which a shared secret is decided by two (or more) parties, as a work of the information contributed by both parties, such that no one is able to predetermine incoming value. [1]

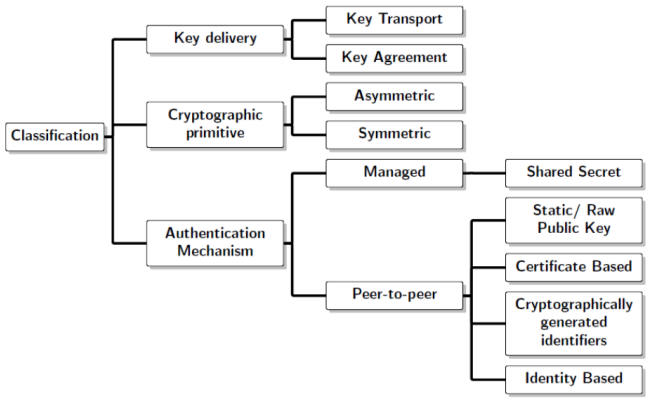


Fig. 1. Classification of Bootstrapping Methods [1]

### B. Cryptographic Approach

In the context of the cryptographic approach, key bootstrapping approaches are classified on cryptography primitive family and depending on fundamental cryptographic primitives can be further classified into symmetric-key schemes and asymmetric key schemes. (i) In Symmetric-key schemes communicating parities have shared a common secret key to encrypt and decrypt messages. It has low overhead and is suitable for constrained- nodes but a raised issue of how to safely pre-configure or transmit keys to the nodes. (ii) Asymmetric key schemes are based on Public Key Cryptography (PKC) which works on the concept of key pairs, the public key, and the private key. Public key of an entity is known to all other entities who want to communicate and the private key is kept secret by the entity. Certainly, PKC is best available security mechanism but due to high power and computational needs it does not suit very well in constrained- node network in standard form. [1]

### C. Authentication Method

This classification of bootstrapping approaches is based on authentication mechanism which is used to identify the users and devices. It can be managed like pre-shared secret keys or peer-to-peer certificate based approaches. [1]

## III. Key Bootstrapping Protocols

### A. PKI based Approach

Public-key cryptography is the current standard for encryption and authentication to achieve security in the communication channels. It is one of the most secure approaches known till date and widely practiced in the industry. PKI provides trust services, namely Confidentiality, Integrity, and Authenticity. It defines policies and practices to manage public-key cryptography and digital certificate creation and its management. In PKI, CA (Certificate Authority) signs and issues certificates to entities, RA (Registration Authority) verifies the identity and ensures registration, a Repository that stores certificates and CRL(Certificate Revocation Lists). Certificate issued by CA binds entity identification to its public key which is part of the certificate. In this section, we will be mainly focused on the PKI approaches/models for the constrained-node network. [1]

*1) **Implicit Certificate Based**:* Implicit certificates are the special variant of explicit certificate where public key, identification data and digital signature is superimposed to reduce the size of the whole certificate to the size of public key. This makes implicit certificates very small compared to explicit certificates. In this method, the public key can be extracted and verified from the signature part of the digital certificate, which is used to extract the public key and use it for operation without explicitly validating the signature of the CA. This makes it very attractive for IoT devices and applications due to its small size and lesser processing. [4]

Two-Phase Authentication solution is proposed by Pawani Porambage et al. which is inspired by ECQV (Elliptic Curve Qu-Vanstone) and ECDH key exchange mechanism. Method has two phases, Registration Phase which mainly deals with obtaining security credentials from Certificate Authority (CA) and Authentication Phase which defines how to start trusted communication between two entities in a network using credentials obtained in earlier phase.

In the registration phase as shown in Fig. 2, node/edge devices request security credentials from the Certificate Authority (CA) which issues implicit certificates once request is received and requester identity is established. Node starts this handshake by sending *Requester Hello, Node Identity and Cipher Suits* that are supported by requester. CA verifies the legitimacy of the node by their identity, on successful validation CA responds with *Hello* message with its public key. Node, on receipt of response, generates node certificate request EC point, true nonce, and calculates MAC (Message Authentication Code) and sends Certificate Request to CA. CA verifies MAC received and generates implicit certificate and private construction key and sends Certificate including nonce and MAC to node. Node receives a certificate and computes its private and public key. Node initiates a finish message to CA to finish the registration process by sending an encrypted digest of previous handshake, for which CA also responds with a finish message to complete the registration process. [5]

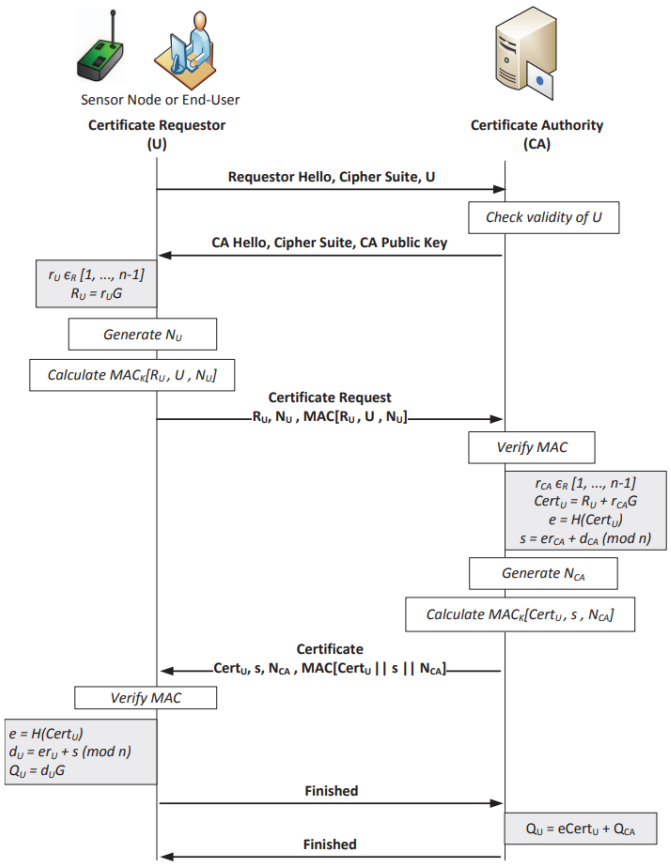


Fig. 2. Registration Phases [5]

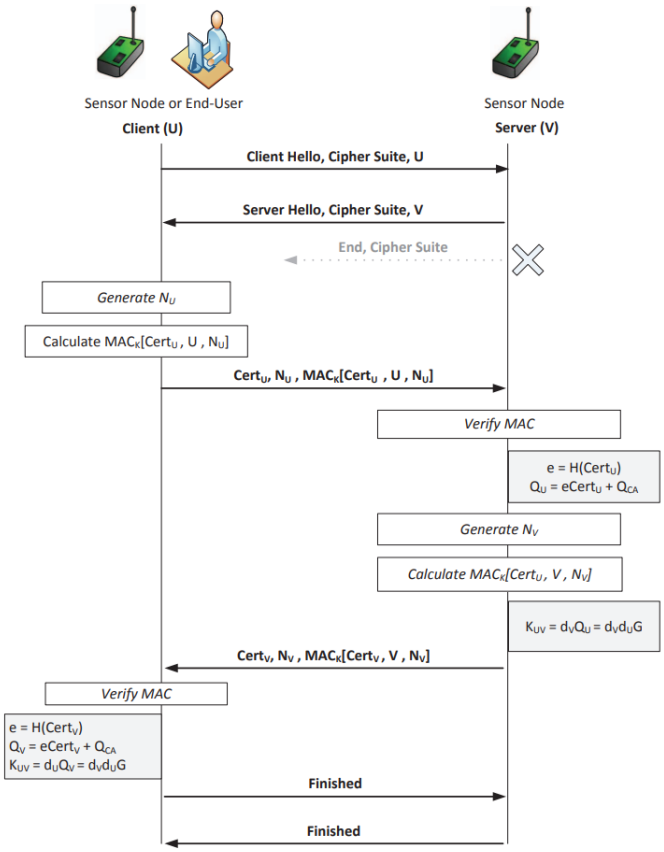


Fig. 3. Authentication Phases [5]

In the Authentication Phase as shown in Fig. 3, nodes/clients send the Hello message with its identity and cipher suite to the server. Now, the important part here is that the client only sends the cipher suites which its implicit certificate is composed of and if the server has a certificate which matches the given list of cipher suites, it agrees to one cipher suite and responds with Server Hello and its identity. If the server finds no match for the cipher suite then it ends the handshake by sending End Message. Once Server Hello is received to the node, it sends its certificate, nonce, and MAC value to the server. Server calculates the client's public key and further calculates the common key based on its private key and client public key and sends its certificate, nonce and MAC as response to the client. Client derives a common key using its own private key and public key of the server. Phase ends with a Finish message like in earlier phases. [5]

Author concludes that after these steps finish, nodes can identify each other and communicate over a secure channel. Author also emphasize that this technique needs a strong identification mechanism, which is secured. Node capture attacks might be successful with weak identification mechanisms and this problem is not addressed in this proposed solution. [5]

Another scheme which is also based on Implicit Certificate is proposed by Mahmud Hossain et al. which provides a lightweight mutual authentication scheme ensuring privacy-preserving identity usage. Proposed scheme authenticates IoT devices without disclosing their identities. This is a two phase security providing scheme namely Network Phase and Service Phase. Scheme is based on Elliptic Curve Qu-Venstone where the device generates One Time Device Identity (OTDI) using Combined Public Key (CPK) cryptosystem and the device is issued a temporal Device ID after verification to join the network. Basic idea of solution and important terms are explained below.

QR Code- Generated and attached to device by manufacturer which is scanned by the application provided by manufacturer for device registration. It contains a URL for device registration, device identity provider, one time cartographic key. This key is encrypted using one-time pad encryption. Device Registration Service (DRS) - DRS stores serial number and Hash of the device which is used to decrypt one time key. Device Identity Provider (DIP) - DIP Maintains a three dimensional public key matrix (PKM). It is the public key of the ECC pair matrix. DIP stores hash of serial number of device serial number and common one time key key. Mobile Application - Application reads QR code and retrieves encrypted key and other details and passes to DIP securely. Network Access Service (NAS) - NAS verifies device identities, issues implicit

certificates. It interacts with DIP to verify the identity of devices.

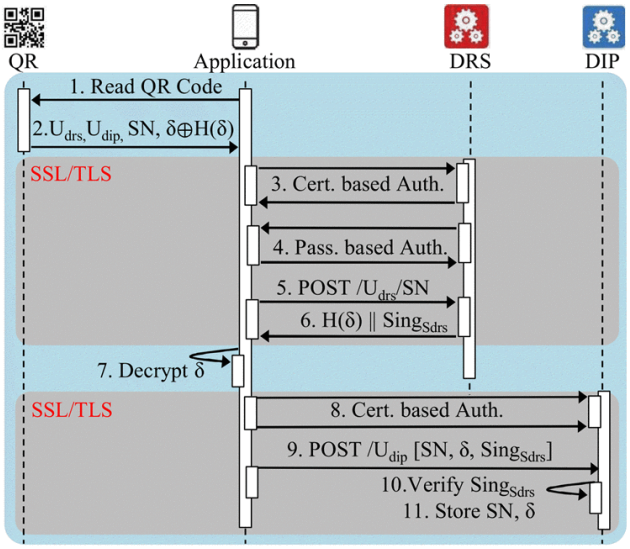


Fig. 4. Device Registration [6]

In the Device Registration Phase Mobile application reads and retrieves all the data including one time key and sends it to DRS over a secure channel. DRS in turns responds to applications with the signature and marks SN as used, making sure not to register again in future. Application decrypts the key received and performs certificates based authentication with DIP and establishes a secure channel. Applications send device serial number SN, hash and signature to DIP which is verified. DIP calculates hash of device SN and stores it in a database along with a one time cryptographic key. Now, device and DIP share a key which will be used to authenticate a device during enrollment.

In the Enrollment phase, DIP and device authenticate each other by one-time shared key. In this phase, the device computes its hash and Message Authentication Code (MAC) and sends it to DIP. DIP retrieves the key from the database and verifies and authenticates the device. DIP uses defined algorithms to authenticate devices. DIP sends MAC, id and other details to the device after verification which is verified on device side and authenticates the DIP. Device encrypted public key and sent to DIP which is decrypted by DIP and stored. DIP sends the public key to its ECC pair for further communications. Device and DIP concludes this phase by destroying the common shared key. In the Network Access, device joining the network computes a one time ECC pair in such a way that it is not been used before. Device sends network access request to NAS, after verification and receipt of device MAC, NAS and DIP performs certificate based mutual authentication. DIP authenticates the device by matching ECC pair and send computed MAC and nonce to NAS. Similarly, NAS authenticates smart device and smart device authenticates NAS and DIP.

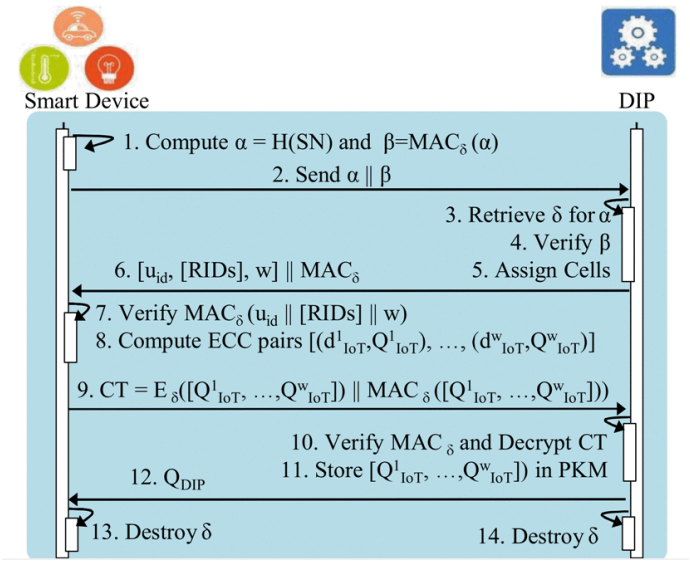


Fig. 5. Enrollment [6]

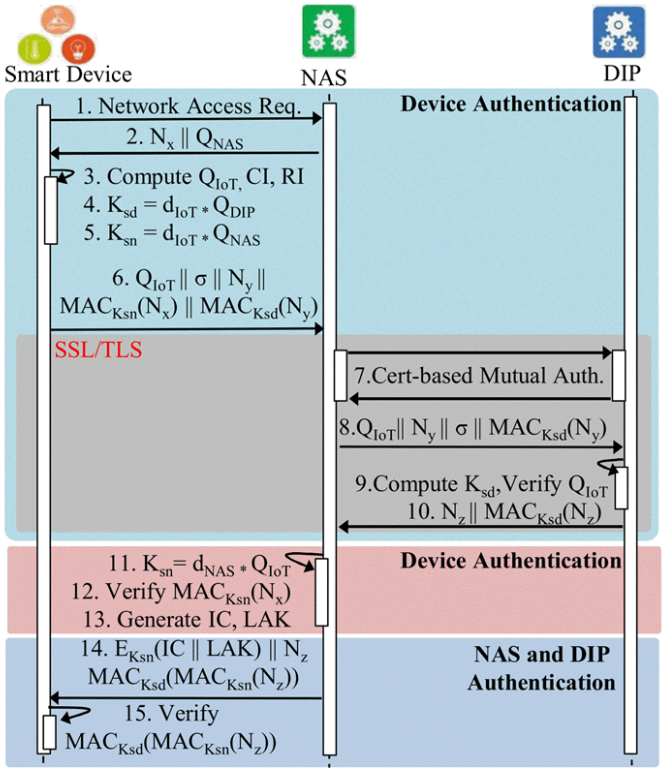


Fig. 6. Network Access [6]

This way, author claims that every entity authenticates each other in the solution and as hash is not computed in the signing process and hash of ICs for authentication, it unburdens constrained devices from expensive computation of hash functions. [6]

### B. *Certificateless Approach*

Secure communication is a very important part of IoT, for data integrity and identity authentication certificateless schemes are used, which is a feasible cryptographic tool to eliminate the escrow problem and complicated certificate management in the certificate-based scheme which is no longer suitable for resource-constrained IoT environments.

The certificateless scheme is based on the intractability of the ECDLP (Elliptic Curve Discrete Logarithm problem). Below, Fig. 4, is one of the general concepts of certificateless scheme involving six steps *Setup, PartialPrivateKeyExtract, SetSecretValue, SetPublicKey, Sign, and Verify*. Out of this six steps, four steps are *Setup, PartialPrivateKeyExtract, Set SecretValue and SetPublicKey* can be treated as preprocessing scheme. Author *Kuo-Hui Yeh et al*, proposed a new connectionless signature scheme based on ECC point-based crypto-operations. Proposed scheme involves two phases, Preprocessing phase and Sign/Verify phase. In the scheme KGC (trusted Key Generation Center), the signer and the verifier are involved. [7]

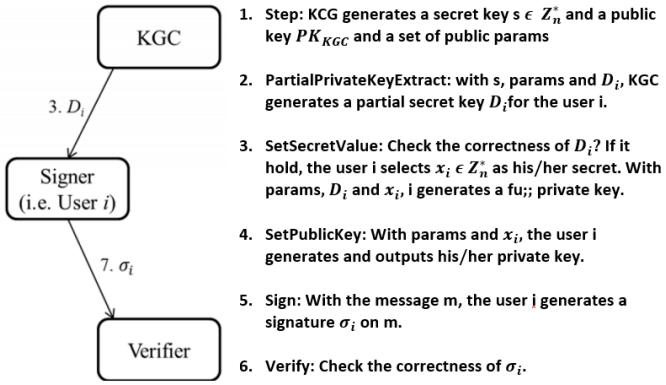


Fig. 7. General certificateless signature scheme [7]

In the Pre-Processing phase Fig. 5, KGC generates a group of elliptic curve points with prime order and determines a generator. Then, KGC chooses a master key and a secure hash function after which KGC calculates a master public key. KGC publishes params and keeps the master key securely. With generated params and identity of the user KGC generates a random number. [7] KGC returns a calculated partial private

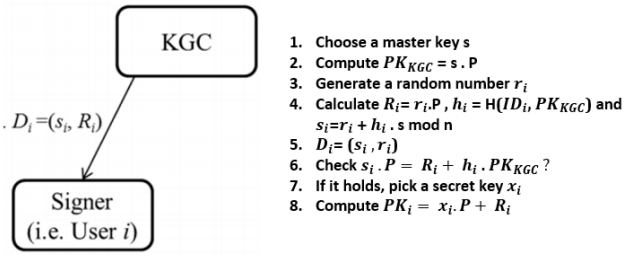


Fig. 8. Pre-processing phase of the proposed certificateless scheme [7]

key to the user who validates the key and checks for the correctness of the key by mathematically checking the equation. On successful validation, the user chooses a random number as own secret value and with the available params, generates a public key. [7]

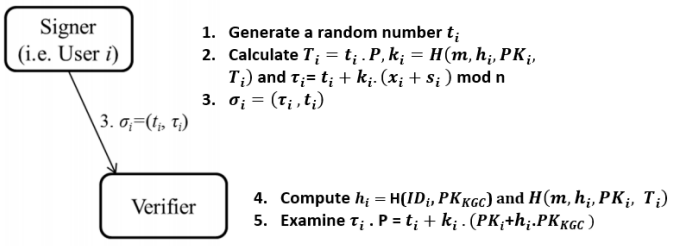


Fig. 9. Authentication Phases [7]

In the Sign/Verify phase Fig. 6, the user generates a random number with the given params and message and calculates the final signature of the message. This is part of Signer and this signature is sent to the verifier who computes and verifies the calculations based on signature received and if successful then signature is accepted.

Author also suggested using a proposed certificateless signature scheme with at least 384-Bit elliptic curve and SHA-3 to achieve highest security. [7]

## IV. Key Management Schemes

To achieve security, encryption and authentication of messages sent between communicating entities must be established. To achieve this keys for performing encryption must be agreed upon by the communicating parties. How to set up these secret keys is known as *Key agreement* problem. There are several key management schemes available and proposed but not all fit into the IoT environment of constrained nodes. Generally, key management schemes can be classified into three types. *(i) Trusted-server scheme (ii) Key Pre-distribution scheme (iii) Public-key schemes* [8]

In this paper, we will be only focusing on those schemes which are relevant to constrained IoT networks/devices.

### A. *Random Key Pre-Distribution Scheme*

A random key pre-distribution approach was given in [9]. This is a three phase key distribution approach which involves pre-distribution, shared-key discovery and path-key establishment. In the first phase a large pool of P keys and their key identifiers are generated. Out of these P keys k keys are drawn to establish the key ring of sensors and the key ring is loaded into memory of each sensor. The second phase, which is the shared-key discovery phase, takes place during DNS initialization in the operational environment where nodes coordinate with each other and discover the same secret key between nodes. This phase establishes the topology of the sensor array and a link exists between two sensors only if they share a key. The path-key establishment phase is used when a common key is not found. Path-key is assigned to nodes which do not share a common key but are connected to two or more links at the end of the second phase. Approach

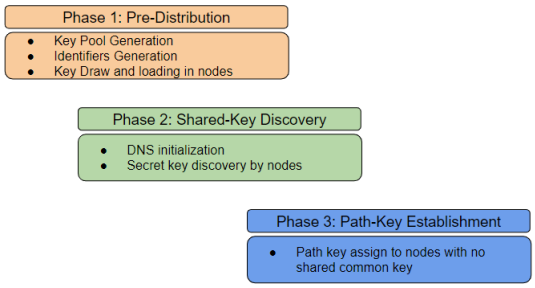


Fig. 10. Random Key Pre-Distribution Scheme Phases

also defines revocation of key ring of compromised node in case of attack. [9] This approach does not provide surety for availability of shared key between pair of nodes but provides guarantee problematically. [8]

### *B. Self-Certified Key Management Scheme*

This self-certified key management scheme is based on a heterogeneous framework in which a network consists of a number of sensor nodes and base station that controls the network and collects data from the nodes. Sensor nodes are connected to the base station and the authentication table of valid sensor nodes is maintained at the base station. Adopting a self-certified public key system approach has three main features. (i) Node and base station can determine the secret key of node. (ii) Node's own secret key can be used to verify the authenticity of the self-certified certificate issued by the base station, avoiding the high-cost public key infrastructure as no extra certificate is needed. (iii) A cryptographic application can be used next to do the public key verification task. [10]

Authentication phase involves mainly two phases in this scheme, the Registration Phase and the Session Phase.

In the registration phase member nodes send a registration request(RRQ) which contains hashed master key and identity id to register member node keys. On receiving the request Base Station validates and recomputes the ID of the node from the has and generates member node's public key which is sent to member node. [10]

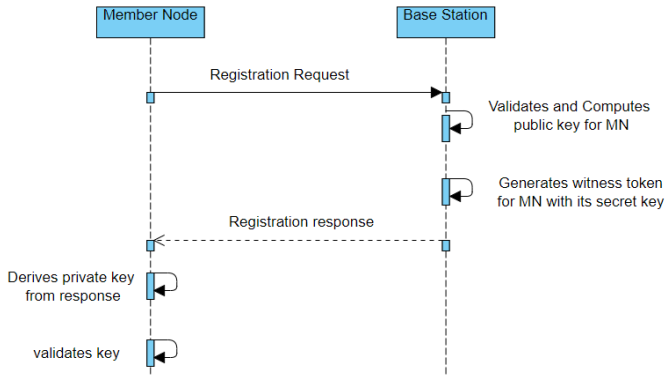


Fig. 11. Registration Phase

In the session phase, the member node sends a session request(SRQ) to the base station which consists of a Time Token and session random number. Base station generates Time token, Session token and session key when request for SRQ is received. BS sends session token to member node which in turns generates session key from session token. [10]

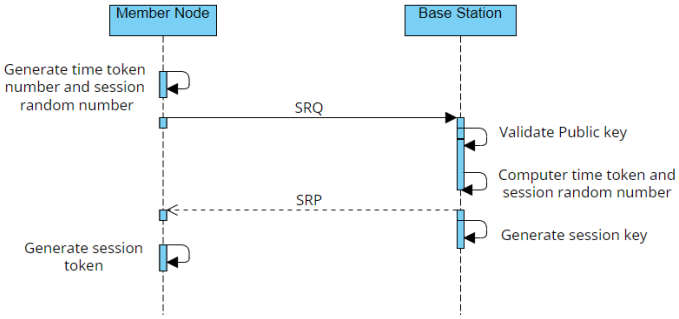


Fig. 12. Session Phase

Author proposes this ECC based light weight self-certified key management and mutual authentication scheme for constrained devices also proves that this method successfully applies zero knowledge technique in his study. [10]

## V. Conclusion

In this paper, we reviewed various existing approaches to achieve security during bootstrapping of constrained-node network and also the key management schemes. We looked how key bootstrapping is effective to make sure communications are secure and not one approach fits everywhere and we need different solutions to solve different problems of security. We surveyed bootstrapping approaches and key management scheme of different categories. We looked at approaches which focused on implicit certificates for bootstrapping as well as certificateless approach. Similarly we also explored key management schemes which are based on pre-distribution of keys. We also found schemes that are designed for self-certified keys. After the study, we found that are very limited surveys are available for this special field of secure bootstrapping and key management for constrained node networks.

## References

[1] M. Malik, M. Dutta, and J. Granjal, "A survey of key bootstrapping protocols based on public key cryptography in the internet of things," *IEEE Access*, vol. 7, pp. 27 443–27 464, 2019.

[2] M. S. B. Sarikaya and D. Garcia-Carillo, "Secure iot bootstrapping: A survey (work in progress)," *Internet-Draft draft-sarikaya-t2trgsbootstrapping-05, IETF Secretariat, Sep. 2018.*

[3] "bootstrapping security - capillary networks and constrained devices," *ericsson White paper*. [Online]. Available: https://www.gsma.com/membership/wp-content/uploads/2016/03/wp-iot-security.pdf

[4] Certicom, "Explaining implicit certificates," *Certicom, Mississauga, ON, Canada, Tech. Rep., 2016.*

[5] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "Two-phase authentication protocol for wireless sensor networks in distributed iot applications," in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, 2014, pp. 2728–2733.

[6] M. Hossain and R. Hasan, "Boot-iot: A privacy-aware authentication scheme for secure bootstrapping of iot nodes," in *2017 IEEE International Congress on Internet of Things (ICIOT)*, 2017, pp. 1–8.

[7] C. C. K.-K. C. W. Yeh, K.-H.; Su, "A novel certificateless signature scheme for smart objects in the internet-of-things," *Sensors,2017,17, 1001. https://doi.org/10.3390/s17051001*.

[8] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Trans. Inf. Syst. Secur.*, vol. 8, no. 2, p. 228–258, may 2005. [Online]. Available: https://doi.org/10.1145/1065545.1065548

[9] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," 2002.

[10] A. P. Haripriya and K. Kulothungan, "Ecc based self-certified key management scheme for mutual authentication in internet of things," *International Conference on Emerging Technological Trends (ICETT), 2016*, 2016.