Aalto University
School of Science
Master's Programme in Security and Cloud Computing (SECCLO)

Shamim Biswas

Enhancing the Privacy of Decentralized Identifiers with Ring Signatures

Master's Thesis
Espoo, July 27, 2020

Supervisors:	Professor Raimo Kantola, Aalto University
	Professor Katina Kralevska, Norwegian University of Science and Technology
Advisors:	Yki Kortesniemi D.Sc. (Tech.), Aalto University
	Dmitrij Lagutin D.Sc. (Tech.), Aalto University

Aalto University
School of Science
Master's Programme in Security and Cloud Computing (SECCLO)

ABSTRACT OF MASTER'S THESIS

Author: Shamim Biswas

Title:
Enhancing the Privacy of Decentralized Identifiers with Ring Signatures

Date: July 27, 2020 **Pages:** 76

Major: Security and Cloud Computing **Code:** SCI3084

Supervisors: Professor Raimo Kantola
Professor Katina Kralevska

Advisors: Yki Kortesniemi D.Sc. (Tech.)
Dmitrij Lagutin D.Sc. (Tech.)

Most identifiers used today, such as OpenID Connect, are controlled by third parties, which can track how the identifier is used. To overcome this, self-sovereign identifiers, such as Decentralized Identifiers (DIDs), which are entirely owned and managed by the user, have been developed. However, in some cases even DIDs alone do not sufficiently protect the user's privacy. For example, if a service can be accessed at multiple fixed locations, using the same identifier repeatedly for each location may over time also reveal the user's location. One of the techniques to hide the exact service identifiers are ring signatures, which enable the generation of anonymous signatures where the real signer's identity is hidden in a set of possible signers.

This thesis takes the use case of electric vehicle charging, where the electric vehicle location may be revealed if static identifiers are used by the electric vehicles and charging stations. A previous solution uses a new ephemeral DID for every interaction, but this requires the creation of a large number of DIDs. This thesis examines an alternative approach of using ring signatures to achieve better privacy with a lower number of DIDs.

The major outcomes of this thesis include how to implement ring signatures for anonymous authentication, comparison of resource consumption with respect to the previous solution, and the applicability of ring signature technology on a broader scale such as in constrained devices. The performance of the new solution was compared with the existing solution by implementing prototypes on Android phones, which communicate over Bluetooth. An assumption on the number of charging events was made based on real data for the country of Norway. The results show that ring signatures are easy to implement and provide slightly better privacy but they are significantly more resource-intensive in terms of storage (about 2 times more) and processing (about 9 times slower). Therefore, large scale implementation of ring signatures on the constrained devices is challenging.

Keywords: Decentralized Identifiers, Ring Signatures, Privacy, Electric Vehicle Charging, Internet of Things

Language: English

Acknowledgements

As I write the last document of my masters programme, I reflect on the last 2 years and the amazing experience it has been. I would like to thank the SECCLO programme for providing me with this opportunity. This thesis has been a great learning experience and I am proud of the outcome.

Of course, this thesis would not have been possible without the ample support and guidance from my advisers Yki and Dmitrij. I would like to thank them for their patience in reading the many drafts and for always pointing me to the right direction during the development of this thesis. A big thanks to Antonio for his contribution and guidance in the design and implementation of the solution. I would like to thank my supervisors Prof. Raimo Kantola and Prof. Katina Kralevska for the interesting discussions and for supporting me throughout the process.

During the isolation caused by the pandemic, support from family and friends became even more important. I would like thank colleagues at the badminton group for providing me with the much needed stress release. I thank my friends in Finland — Mohammad, Kidus, Sonika, Bruno, Sati, Jim, as well as my friends in India for encouraging me to keep pushing through the challenging times. Last but not the least, I would thank my family — Mom, Dad, Sister, and Brother-in-law for sending their love and care from afar.

Espoo, July 27, 2020

Shamim Biswas

Contents

Abbreviations and Acronyms

BLE	Bluetooth Low Energy
CS	Charging Station
CSO	Charging Station Owner
DID	Decentralised Identifier
DSO	Distribution System Operator
ER	Energy Retailer
EV	Electric Vehicle
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
EVU	Electric Vehicle User
IoT	Internet-of-Things
JSON	Javascript Object Notation
JWS	JSON Web Signature
MitM	Man-in-the-Middle
NFC	Near Field Communication
PKI	Public Key Infrastructure
RSA	Rivest, Shamir & Adleman (public key cryptosystem)
SSI	Self-Sovereign Identity
TSO	Transmission System Operator
UUID	Universally Unique Identifier
VC	Verifiable Credential
ZKP	Zero-Knowledge Proof

Chapter 1

Introduction

Privacy is necessary for an open society in the electronic age. According to Langheinrich [32], privacy is built on the trust that a service respects the user's information. In a privacy protecting service, users know when and what data is being collected and user consent is explicitly required. Additionally, privacy preserving services must not force users for consent to give non-essential personal information in order to use the services such as is the case with many dominant companies like Facebook and Google. Thus, a balance between control and convenience is important.

Digital identifiers are essential for entities to interact with each other, they are used in almost all services for differentiating the users. Digital identifiers, or simply identifiers, uniquely identify an entity among others. An identifier is typically a string of characters, that is uniquely associated with a subset of attributes about an entity. Identifiers are important not only for identification, but also for access control, personalisation and keeping historical information. However, while identifiers are useful for its user, if the user's data leaks into other domains, the same identifier enables correlation and harms user's privacy.

Privacy violations occur when information along with the associated identifier divulged in one context leaks into another. Thus, identifiers should be designed to be privacy protecting. Traditional identifier solutions such as OpenID Connect [51], Shibboleth [14] and Kerberos [58] are centralised solutions which are managed by a third-party authority which may not have user's privacy as their prime interest. The central authority may collect information about the identifier use, such as the services being accessed, time of use, frequency, etc. Identifiers held centrally puts user's privacy at risk as they become attractive targets for hackers. An dishonest authority may even choose to sell those identifiers and the usage data.

That is why self-sovereign identifier systems [62] have been developed

which advocate that identifiers should be owned by the entities using it [53]. Decentralized Identifiers (DIDs) [45] is one such technology which allows entities to create self-sovereign identities, i.e. create and manage their own identity completely independent of any authority. The owner of the identifier has full control over its use, sharing and invalidation. DIDs are designed to be pseudonymous and thus several DIDs can be used simultaneously by a user. Therefore, DIDs enable the user to manage multiple identities with different services without a risk of correlation by someone with access to the interaction data.

However in some cases, more privacy is desired and users wish to authenticate anonymously to the service. These are cases where the user can use one of the several service locations and it is not required by the service provider to exactly know the location where services are accessed. For example, at an automated kiosk machine, the user needs to prove his identity to the kiosk company to get billed and optionally get discounted prices. The kiosk company however does not need to know which kiosk outlet was used for the purchase. Another example could be a door access system to a building. The building security might need to make sure the person accessing the premises has the rights to do so without necessarily knowing the exact gate used for entry. Protection of location information is important as location is very personal information about the user. Location information may be sold for profit to advertisers e.g. location based spamming or it might even be used for more sinister purposes such as tracking, stalking, kidnapping, robbery, etc.

Ring signatures, first proposed by Rivest et. al. [47], is a technique which enables generation of anonymous signatures by hiding the true signer in a set of possible signers. The set of possible signers can be created without the involvement of the other signers and is called a 'ring' due to the shape of the mathematical structure involved. This thesis hypotheses that in situations where it is sufficient to prove that an entity is one of the permitted users of the 'ring', ring signatures can be used as method for anonymous authentication. This thesis also investigates the suitability of ring signatures and the possible privacy improvements using the use case detailed in the next subsection.

1.1 Use Case

This thesis takes the use case of charging of electric vehicles. The use case is adopted from Antonino's thesis [5] which proposes a privacy preserving system for managing identifiers and authentication for *Electric Vehicles* (EVs) and *Charging Stations* (CSs) for a grid balancing electric vehicle charging

scenario. The system aims to balance the grid's energy flow by diverting electric vehicle charging to districts with surplus energy and yet, protect the location privacy of *Electric Vehicle Users* (EVUs). Regulation of energy consumption is required to prevent reverse power flows which can be harmful to the grid [56].

The solution defines the players and information required by each party such that each party has just enough information to allow the energy distributors and sellers to track usage and correctly bill customers without violating the EVU's privacy. The six players associated in the use case are as follows:

1. *Distribution System Operator* (DSO) manages the high-voltage infrastructure transmitting energy from production plants to municipalities and the grid delivering energy to end customers, e.g. houses and small businesses. DSO typically divides its coverage area into different energy districts. Districts facilitate in balancing consumption of energy — surplus energy from one district is routed to energy deficient districts.

2. *Charging Stations* (CSs) are service points where electricity can be accessed from the grid for charging.

3. *Charging Station Owners* (CSOs) are independent entities which own CSs located in one or more districts.

4. *Energy Retailers* (ERs) provide charging services to their customers in partnership with CSO. ERs also fulfil the role of mediators matching the energy needs of the DSO with the charging needs of their customers.

5. *Electric Vehicles* (EVs) interacts with the grid by consuming electricity via CSs.

6. *Electric Vehicle Users* (EVUs) are users of the EVs. EVUs become customers of one or more ERs of their choice to get charging service.

The market relationships are defined as follows and also defined in Figure 1:

- A DSO has agreements with one or more charging station owners (CSOs) which have deployed CSs in one or more energy districts.

- A DSO has agreements with one or more ERs to allow them to sell energy to their customers.

- EVUs are customers of ER. An ER has contracts with several different customer EVUs, and each EVU can be a customer of one or more ERs.

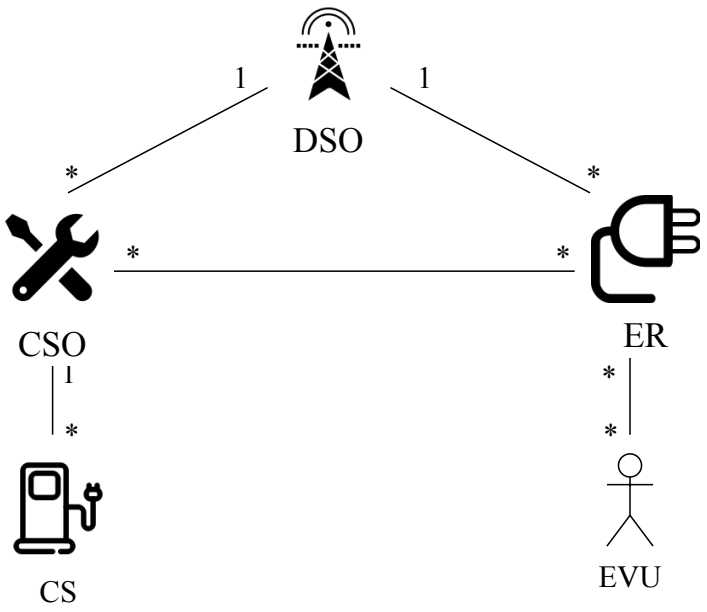

Figure 1.1: Entity-relationship scheme showing the different relationships among the players in the market. [5]

- An ER can provide its services through the CSs belonging to one or more CSOs.
- The CSs of a CSO can be used by one or more ERs.

The payment relationship dictates the amount of information each party actually requires. Every time an EV charges at a charging station (CS), a charging event takes place which produces details such as the specific CS and the EV involved as a transaction log. The ERs bill the EVUs at the end of the billing period and pay the CSOs based on the amount of energy charged through the CSs. Thus, ERs must be able to demand payment from the EVUs, and CSOs must be able to claim payments from ERs by proving the authenticity of the charging events. The charging event authenticity must also be proved to the DSO for ER to claim reward for its grid balancing efforts. The payment resolution is achieved through the transaction logs.

However, the the transaction log reveals the CS identifier to all parties which is a threat to the location privacy of the EVUs. A district-level granularity for charging events is sufficient for the DSO to verify that the customers of a specific ER have charged for a certain amount of energy within that district. Similarly, the ER needs only information about the CSO, for billing purposes, without the need to identify the specific CS that took part

in the charging event. Finally, CSOs only need to authenticate that EVUs are customers of the ER and do not require the EVU's identity.

Therefore, the privacy protection of the EVU in the use case is a double problem. Firstly, the EV must authenticate anonymously at the CS to protect its location privacy from the CSO. And secondly, the CS must authenticate itself anonymously to the ER as otherwise the CS identity, and therefore also the location of the EVU, is revealed to the ER through the transaction logs. The proposed system by Antonino [5] created different DIDs for both the EVs and CSs for each charging transaction to protect the EVU privacy. However, this approach leads to creation and storage of a large number of DIDs and correspondingly increases the amount of resource consumption.

1.2 Scope and Research Questions

This thesis focuses on the interaction between the EV and CS during the charging event and the payment resolution between the CSO, ER and DSO with the aim to satisfy all the privacy requirements discussed before and yet provide sufficient billing evidence. To protect the privacy of the EVU, this thesis designs an alternate solution using ring signature. It investigates the suitability of using ring signatures for the CSs to sign the transactions instead of creating new CS identifiers (DIDs) at every charging event. As the ER and DSO should be able to identify the location of a charging event at the district level instead of the exact location, the charging data could be signed with a ring signature containing all the charging stations in that district as the set of possible signers. A ring signature could be created by any CS in a district but the signature would not reveal the identity of the real signer. A verifier can then verify that the transaction happened in some CS in that district without knowing the exact CS identity. The relative performance of ring signatures with respect to using just DIDs is also compared by simulating the charging event on mobile devices (Android smartphones). More precisely, the thesis answers the following research questions:

1. **RQ1**: How should the charging events be logged and signed with Ring Signatures without revealing personal information of the user and yet prove the authenticity of the charging transaction to the CSO, ER, and DSO?

2. **RQ2**: How does the use of Ring Signatures along with Decentralized Identifiers, as opposed to using just Decentralized Identifiers, affect the protection of privacy for EVUs in the use case?

3. **RQ3**: What is the effect of using Ring Signatures on the resource consumption and the transaction time of the system in the use case?
4. **RQ4**: What effect do Ring Signatures have on the deployability of the system on constrained devices?
5. **RQ5**: How do Ring Signatures affect the use of Decentralized Identifiers on a broader scale?

1.3 Structure

The rest of the document is structured as follows. Chapter 2 introduces the key technologies including Decentralized Identifiers and Ring Signatures and the related research in the field of privacy protection as well as describes the baseline design used for comparison in the thesis. Chapter 3 describes the design choices and the modified design. Chapter 4 provides implementation details of both the baseline and modified design. Chapter 5 presents the results from the experiments and its analysis. Chapter 6 addresses the research questions and discusses the outcome of the study. Chapter 7 presents future work. And finally, Chapter 8 concludes the thesis.

Chapter 2

Related Work

This section presents the related work done in the field of privacy in IoT interactions. First, it explains the key technologies used in this thesis such as Decentralized Identifiers (DIDs), Verifiable Credentials (VCs) and Ring Signatures. Then, it presents privacy enhancing techniques developed for IoT Devices. Next, it describes the location privacy preservation techniques specific to electric vehicle charging. After that, it presents the privacy preserving system developed by Antonino [5] for a grid balancing electric vehicle charging scenario . And lastly, it describes the improved electric vehicle charging design developed by Antonino et. al. [6] which is used as the baseline design in this thesis for comparison.

2.1 Decentralized Identifiers (DIDs)

Decentralized Identifiers (DIDs) [45] are a self-sovereign identifier technology being developed by the World Wide Web Consortium (W3C). The major advantage of DIDs is that the DID infrastructure is decoupled from a single authority and therefore DIDs provide more privacy and control to the users by allowing to create, manage and destroy DIDs without interference from another party. DIDs are based on public key cryptography and one or more key-pairs can be generated by the user for a DID. DIDs are extensible and many standards and versions of DIDs exist, but broadly they are of two types — public DID and peer DID.

Public DIDs are known publicly available and are meant to used by multiple entities to communicate with the DID owner. The keys associated with public DIDs are made public via writing on a distributed ledger, a Domain Name System (DNS) or a website. Distributed ledgers can act as a Public Key Infrastructure (PKI), adding trust to a public DID as legitimate iden-

tifier of public identities such as institutions. Distributed ledgers may be implemented as one of the several types of blockchain instances available such as Sovrin [62] which uses its own identity blockchain, and uPort [35] which uses the Ethereum network [65]. Newer blockchain technologies have more efficient cryptographic properties [44] and have the potential to provide consensus with less resource consumption.

However, public DIDs are unsuitable for use as ephemeral identifiers i.e. using identifiers for just one interaction. Writing the DID and its associated key to ledger is not only unnecessary but also a risk for privacy as the keys are required to be known only to the entity being communicated with. Writing to public ledgers usually costs money and it becomes expensive and wasteful when DIDs usage is short-lived and not even required to be publicly accessible. Furthermore, resolving a key associated with a public DID is slower than resolving peer DIDs due to the network overhead from reading the ledger.

On the other hand, peer DIDs [42] are not required to be written on the ledger. Peer DIDs are free from this requirement as they encode their keys into the DID itself making the keys readily available to the communicating party. Therefore, peer DIDs are suitable for use as ephemeral identifiers.

All kinds of DIDs and the associated secret keys are stored securely in identity wallets [25]. Identity wallets prevent leak of secret keys by allowing the use of secret keys eg. signing, without bringing the keys out of the wallet. An identity wallet is accessed with a master password known only to the wallet owner.

2.2 Verifiable Credentials (VCs)

Verifiable Credentials (VCs) [55] are another technology being developed by W3C which complements DIDs in creating a decentralised and verifiable identifier system. A verifiable credential is a signed document containing a set of claims about a subject. VCs are digital world equivalents of physical proofs such as drivers licence, birth certificate, etc. VCs, like DIDs, can be stored securely in identity wallets. VCs facilitate making claims with the help of proofs created by the issuer. There are different types of proofs which can be used in VCs [20] such as Zero Knowledge Proof (ZKP) and JSON Web Signature (JWS) [29]. The ZKPs are complex structures but provide advanced properties such as anonymous presentation of the credential and selective disclosure of claims. The JWS type verifiable credentials are cryptographically much simpler and therefore are suitable for making simple claims. One of the signature schemes recently developed for elliptic key cryptography is

the JCS Ed25519 Signature 2020 [16]. The JCS Ed25519 Signature 2020 normalises the VC data using the JSON Canonicalization Scheme (JCS) [49] and creates signatures using the Ed25519 signature algorithm [10].

One of the ways to represent VCs is using linked data structures [19] such as JSON-LD [54], which is a lightweight syntax to serialise Linked Data into JSON. The credentials consist of context, subject and the proof. The context contains a URI which links to a documents explaining all the fields and usage and policies of the credential. The subject is the data which is authorised such as DID or other claims. The proof contains the signature of the credential issuer and other information about the signature such as algorithm, timestamp and verification methods.

2.3 Ring Signatures

Ring signatures are a technique to create anonymous signatures. They were first proposed by Rivest [47] and then redefined by Bender [8]. Ring signatures hide the identity of the real signer in a set of possible signers called the 'ring'. The number of signers in ring is called the ring size. The ring formation does not require the participation or permission from the other signers. The generation of a ring signature involves first choosing random values for each of the non signers and then finding the correct value for the signer to satisfy a condition. The real signer is able to do find such a value as it knows the secret key. The detailed construction process of ring signatures is complex and therefore is not described here. Readers are encouraged read the construction in the original paper [47]. Other variants of ring signatures have also been developed such as threshold ring signatures and linkable ring signatures [40]. Ring Signatures can be created using any signing algorithm such as RSA or Edwards-curve Digital Signature Algorithm (EdDSA) [52]. This thesis uses the Ed25519 signing algorithm [10], a type of EdDSA, which is based on the Curve25519 [9].

2.4 Enhancing Privacy in IoT Devices

An analysis of identifiers in popular IoT platforms including oneM2M [60] and FIWARE [15] has been done by H. Aftab et al. [2]. However, adopting a resource based approach to identification, they use static identifiers and do not analyse privacy threats to using static identifiers.

While communicating with an external party, IoT devices need to use pseudonymous identifiers to hide their identity. However, when the same

identifier is used, over time it can be correlated and mined for behavioural pattern. To address this issue, the use of new identifier for each interaction has been suggested by Kortesniemi et al. [30].

The use of DIDs for securing IoT device registration and software update in 5G has been shown by Ansey et. al. [4]. An attempt to decentralise OpenId Connect by using distributed ledger backed DIDs and VCs has been presented by Lux et al. [36]. DID is a new technology and hence there is limited work available which optimise their use for services.

Bender et al. [8] suggest the use of ring signatures to provide a member of a certain class of users access to a particular resource without explicitly identifying this member. Ring signature can provide undeniable proof that such a resource has been accessed. The use of ring signature for preserving privacy in digital transactions such as e-payments and e-voting services has been proposed by Malina et al. [37]. While their design provides authentication and anonymity within the group, it does not specify any method to prove authentication to a third party. Improved ring signatures using the Rabin cryptosystem has been proposed but not yet implemented [37].

In comparison, work by Yang et al. [66] presents the use of Zero Knowledge Proofs (ZKP) as anonymous credentials for anonymous authentication but lacks a full communication protocol. A privacy preserving authentication protocol for IoT devices has been proposed by Wang [63], but since the protocol uses group signatures, it requires additional steps such as group creation, key distribution and expensive cryptographic operations.

2.5 Location Privacy of Electric Vehicles (EVs)

The state of art EV charging infrastructure involves EVUs using ER specific mobile application, and authenticating themselves with registered username and password, or alternatively using ER issued RFID tags. Both the mechanisms use static identifiers that reveal EVU's location to the CSO and the ER. Furthermore, payments are usually done via credit card which again unnecessarily reveals EVU information [24].

Location privacy protection for EVs is a much researched field and several privacy preserving methods have been proposed. Au et al. [7] have developed a privacy preserving payment scheme using Zero Knowledge Proofs (ZKPs) for transactions and stored balance accounts for payments. However, since the cryptographic operations involved in ZKP are intensive, CS operations are offloaded to a billing server. Other methods use a trusted third-party which performs aggregation of data from geographically co-located consumers so that service provider can still get an aggregate view of the transaction

events for a street or district, but not for a single customer [31]. The reader is encouraged to refer to Antonino's thesis [5] which provides extensive related work in this field (in Section 3) such as the use of escrow for safe payment schemes and the use of group and ring signatures and hence has not been repeated here.

2.6 Privacy Preserving and Grid Balancing EV Charging

Antonino's privacy preserving system [5] for a grid balancing electric vehicle charging scenario which was introduced in Section 1.1 is explained in more details here. The system uses different DIDs and VCs for the EV and CS for each charging transaction. DIDs are also used for the credential issuers i.e. the *Charging Station Owner* (CSO) and the *Energy Retailer* (ER). The charging transactions are signed by the EV and are later used for payment resolution by the CSO, ER and the DSO at the end of the billing cycle. The service location is divided into districts by the *Distribution System Operator* (DSO) and the electric grid is balanced by scheduling charging events to district with power surplus. Scheduling is done using economic incentives. The DSO rewards the ER for consuming surplus electricity in a district in a given time period and the ER offers discounts to EVUs to encourage charging in the target district. The location privacy of the EVU is protected by the division of knowledge between the involved parties i.e. CSO, ER and DSO as shown in Figure 2.1. Following the principle of least information, none of the parties has enough information to identify the EVU location provided there is no collusion between the parties. The CSO knows the location of the transaction but knows only anonymous identifier of the EVU (its DID). Similarly, the ER knows the EVU identity but only knows the anonymous identifier of the CS. The DSO does not know the identity of both the CS and the EV. The system does not provide implementation details but suggested the use of Hyperledger Indy [34] for DID and VC creation.

2.7 Multi-DID Design for Privacy Preserving EV Charging

Starting from the design laid down in Antonino's thesis [5], an improved design has been developed jointly with Antonino et al. [6], henceforth referred to as the Multi-DID Design. The original design has been refined by

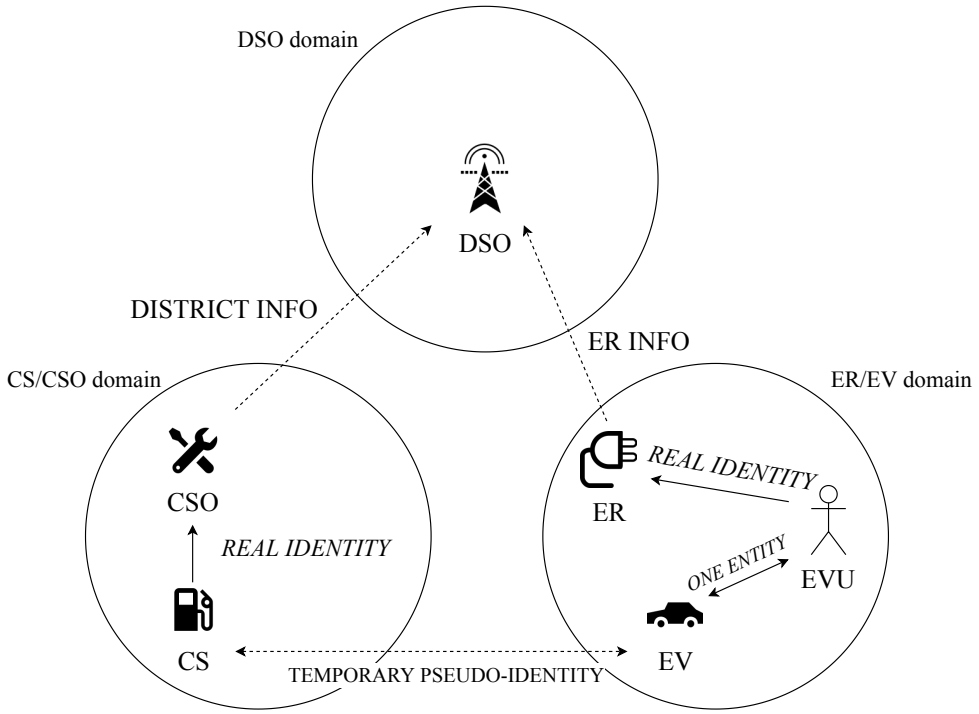

Figure 2.1: The different knowledge domains of the use case described. Solid arrows indicate personally identifiable information. Dashed arrows indicate information that do not relate to a single entity (either CS or EV) [5]

adding more details and made more efficient and secure by incorporating Peer DID [42], DID Exchange Protocol [50], Linked Data Proofs [19], hash chain based micropayments [46], and credential reorganisation.

2.7.1 Credential Generation

The Multi-DID Design uses two kinds of credentials for authorisation of the DIDs used by the EV and CS. Firstly, the credential used by the EV is called the EV Credential. The EVU registers itself as a customer with one or more ERs as a customer to be able to charge their EV. The ER may verify the EV and EVU using its own authentication mechanism. Before using the charging event, the EV generates DIDs, henceforth called EV DIDs, and sends them to the ER. The ER issues credentials to the EV, where the subject of the credential is a single EV DID. The EV Credential validates the owner of the EV DID for charging at CSs which have an agreement to provide service for the ER. At the end of the billing period, the EV Credentials enable the verification of transaction logs to the CSO and the DSO. The EV Credential

can be verified using the public DID of the ER denoted as ER DID. The associated public key of ER DID is resolved on the distributed ledger. Since EV DID should not be reused to protect the privacy of EVU, EV can request as many credentials as needed (up to a limit) from the ER. The EV Credential is valid for 5 days.

Secondly, the credential used by the CS is called the CS Credential. Each day, the CS creates as many DIDs as the number of charging transactions expected and forwards the DIDs to the CSO. These DIDs are called CS DIDs and in order to protect the EVU's location privacy, the CS must use a CS DID only once. The district where the CS is located in is known by the CSO and is denoted by the district ID. The CSO issues CS Credentials which have a single CS DID and the District ID as the subject. The CSO should provide the correct District ID as the DSO has smart meters installed at the CSs from which energy consumption is monitored. If a CSO issues credentials with false District IDs, the total electricity consumed readings from the smart meters will not match with the total consumption from the transaction details and his fraud will be caught. Just before the charging event, the CS Credential enables a CS to authorise itself to the EV as a valid charging station. At the end of billing period, the CS Credential proves the authenticity of the transaction log to the ER and the DSO. The CS Credential can be verified using the public DID of the CSO denoted as CSO DID. The associated public key of CSO DID is also resolved on the distributed ledger. The CS Credential is valid for 3 days.

2.7.2 Exchange Messages and Presentations

The EV and CS communicate over a wireless channel such as Bluetooth during the charging event using peer DIDs. The DID Exchange Protocol [50] is used to securely exchange the EV DID and CS DID. The DID Exchange Protocol is a 4-way handshake protocol and involves the messages: exchange invitation, exchange request, exchange response, and exchange complete. The exchange invitation is the only message which is sent unencrypted and provides a public key to bootstrap the communication. With the exchange request and exchange response messages the communication parties exchange their peer DIDs with each other. Finally, the exchange complete message confirms the DID exchange. The exchange messages between EV and CS are shown in Figure 2.2.

Along with the exchange response and exchange complete messages, credentials and presentations are also sent. Presentations are proof of credential ownership. The presentation consists of two fields: the claims and the proof. The proof field in presentation specify the signature algorithm, time

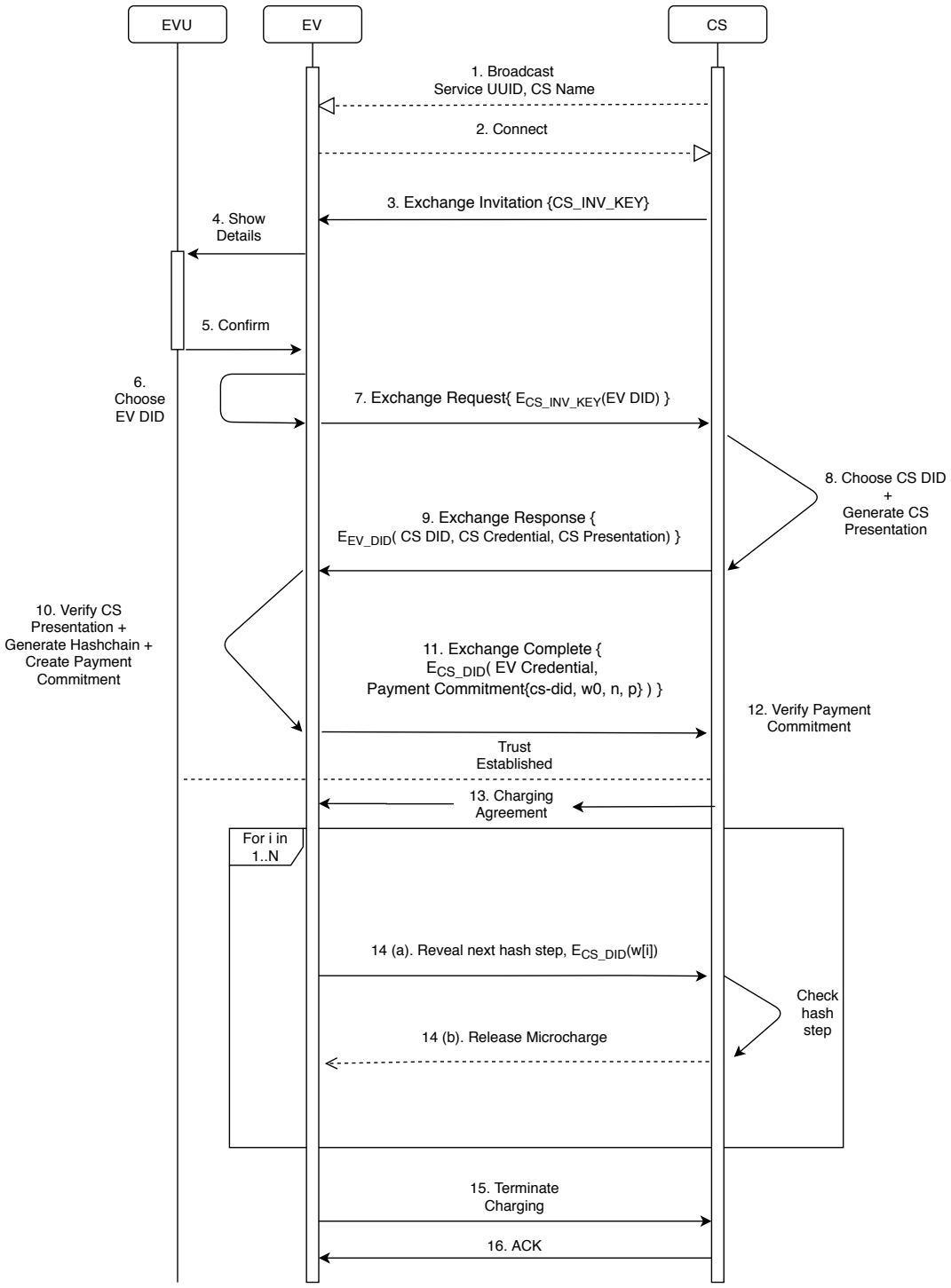

Figure 2.2: Details of Charging Event in reference design

of creation, verification method and the signature value. The signature in a presentation demonstrates the ownership of the peer DID used by the presentation creator. Thus, the EV and CS mutually authenticate each other's peer DIDs by exchanging presentations.

The presentation created by CS is called CS Presentation and it consists of a signature on the EV DID received from EV, created with the secret key associated to the CS DID. The CS Presentation and CS Credential together prove the authenticity of the CS to the EV. On the other hand, the presentation created by the EV is called the Payment Commitment. The Payment Commitment is signed by the EV using the secret key associated with the EV DID. The Payment Commitment and the EV Credential together prove that EV is authorised by the ER to charge.

The Payment Commitment is a proof that the EV has agreed to charge. Once the EV and CS have mutually authenticated each other, the charging of the EV commences using a payment protocol based on the payword micropayments scheme [46]. The micropayments scheme uses hash chains to securely pay a large amount in small increments. The basic idea is that a secret random value is repeatedly hashed 'n' times and each hash value is stored. The last hashed value is called the root of the hash chain 'w_0'. Then, at every hash chain step the pre-image of the last hash is revealed and the CS releases the amount of electric charge corresponding to that step 'p'. This process continues iteratively until the required amount is charged or the maximum length of chain 'n' is reached. Using micropayments minimises financial risk for both parties as the cost of single request not being fulfilled by the CS or inversely, non payment by EV is quite low. Thus, the Payment Commitment consists of the following values: CS DID received from the CS (cs-did), root of the hash chain (w_0), the maximum length of the hash chain (n) and amount of charge to be released per hash chain step (p).

2.7.3 Charging Transaction

The details of the charging transaction in the Multi-DID Design are described here. The steps are indicated in Figure 2.2 and explained as follows:

1. CS broadcasts its availability by advertising a custom service UUID, CS name, and socket number. The service UUID is already known to the EV and the CS name and socket number are printed on the CS. A EV in the vicinity, scans the nearby devices and filters target CSs based on the service UUID, CS name and socket number.

2. EV connects to the CS and preferred connection parameters are negotiated. For the Bluetooth implementation, notification are also need to

be enabled as explained in Chapter 4.

3. CS sends the *Exchange Invitation* after connection establishment to start the communication. The Exchange Invitation contains static public key of the CS, CS_INV_KEY, which is used for encrypted communication with CS.

4. EV shows the discovered CS and its details to EVU.

5. EVU confirms the details of the CS and the charging intent. No actions are required from the EVU after this step.

6. EV chooses one of the EV DIDs from its wallet.

7. EV creates the *Exchange Request* and sends it to CS. The *Exchange Request* is encrypted with CS_INV_KEY and contains the EV DID.

8. CS chooses a CS DID and the corresponding CS Credential from its wallet. CS also generates a CS Presentation by signing the EV DID with private key associated with the CS DID.

9. CS the sends the *Exchange Response* which contains the CS DID, CS Credential, CS Presentation and also the charging options. The charging options are the supported values for max hash chain length and the charge amount for each step. The message in encrypted with public key associated with the EV DID.

10. EV verifies the CS Credential and the CS Presentation. Then, it generates the Payment Commitment. The Payment Commitment is signed with private key associated with the EV DID.

11. EV sends the *Exchange Complete* which contains the EV Credential and the Payment Commitment. The messages is encrypted with public key associated with the CS DID.

12. CS verifies the Payment Commitment after which the EV and CS have established trust with each other.

13. If the CS agrees to the terms in the payment commitment, it sends a charging agreement message to the EV. CS then waits for the reception of the hash chain steps.

14. This step is repeated for the number of hash chain steps used — (a) EV reveals the next hash chain step value. The message is encrypted using public key associated with the CS DID. (b) CS upon receiving

this message, verifies the hash chain step and releases a micro-charge of amount equal to the step charge mentioned in Payment Commitment.

15. After required amount of charging has been done, the EV sends a charging termination request.

16. Upon receiving a termination request, the CS sends an acknowledgement of successful termination.

2.7.4 Payment Resolution

After the charging event, the CS stores the last revealed hash chain value w_k where k is the number of micropayments used. The Payment Commitment, CS Presentation, w_k, and k together are called the *Transaction Log*. The CS sends the Transaction Log along with the EV Credential for each charging event to the CSO. At the end of the billing period, Transaction Log for all charging events in the period and the associated credentials are shared with the ER and DSO as shown in Figure 2.3

1. With the Transaction Log, EV Credential and CS Credential, the CSO is able to:

   (a) Identify the CS involved in the transaction from the CS DID in the Payment Commitment and the district the CS is located in from the CS Credential.

   (b) Identify the ER of which the EVU is a customer and thus demand payment from the ER for the services rendered.

   (c) Identify the amount of energy used for charging from the amount of charge per step 'p' and the number of micropayments 'k'.

   (d) Identify the time of the transaction from the timestamp in the Payment Commitment.

2. Similarly, with the Transaction Log, EV Credential and CS Credential, the ER is able to:

   (a) Identify only the district the transaction happened from the CS Credential. This is because the CS DIDs are different in each transaction and ER does not know the real identity associated with the CS DIDs.

   (b) Identify the CSO whose CS provided the charging service and thus make payments to that CSO appropriately.

(c) Identify the amount of energy used for charging from the amount of charge per step 'p' and the number of micropayments 'k'.

(d) Identify the EVU who made use of the charging service from the EV DID in the EV Credential and appropriately bill them.

(e) Identify the time of the transaction from the timestamp in the Payment Commitment.

3. And finally, with the Transaction Log, EV Credential and CS Credential, the DSO is able to:

(a) Identify the only the district the transaction happened from the CS Credential. This is because the CS DIDs are different in each transaction and DSO does not know the real identity associated with the CS DIDs.

(b) Identify the ER involved in the transaction from the EV Credential and reward the ER appropriately.

(c) Identify the amount of energy used for charging from the amount of charge per step 'p' and the number of micropayments 'k'. The DSO sums the total energy used per district and decides the degree of the ER's contribution to grid balancing.

(d) Identify the time of the transaction from the timestamp in the Payment Commitment.

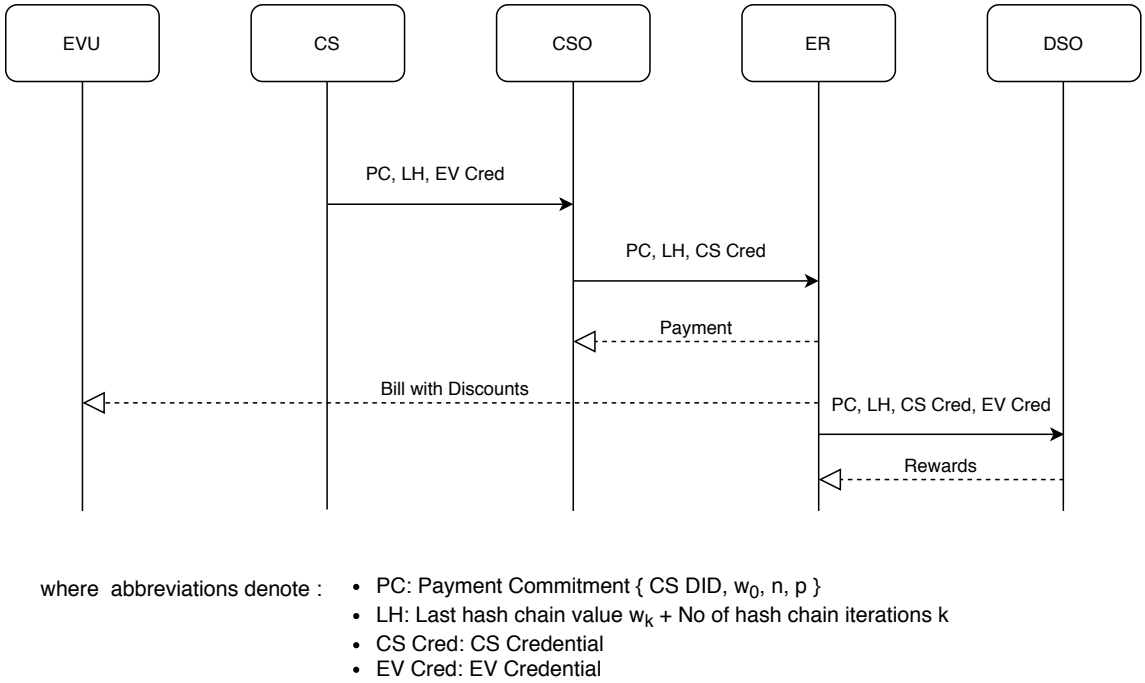

Figure 2.3: Flow of Charging Transaction Logs in Multi-DID Design

Chapter 3

System Design

This section describes the Ring Signature Design which uses ring signatures for privacy preserving EV charging. First, it discusses the design choices made. Then, it describes the assumptions taken for the design. Thereafter, it shows the modifications made to the Multi-DID Design for integrating ring signatures in terms of credential generation, messages and presentations, charging event, and payment resolution. Finally, the limitations of the design are described.

Mainly three changes were done in the Multi-DID design for using ring signature:

1. the CS DID is reused for multiple transactions and used for as long as the CS credential is valid e.g. a month,
2. the CS credential is modified to include CS DIDs belonging to all CSs in the district,
3. and finally, the CS DID in the Payment Commitment is replaced with a ring signature created by the CS.

3.1 Design Choices

This section describes the various design choices made for the Ring Signature Design in terms of identifiers used, type of signatures, credential formats and the threat model.

3.1.1 Identifiers and Ledger

DIDs provide a standard and secure way to communication without interference from a central authority. Public and peer DIDs have different properties

which make them suitable for different operations. Public DIDs, which are useful where long lived relationships are to be maintained, have been used for the service providing entities, i.e., the CSO and the ER. On the other hand, the peer DIDs are suitable for use as ephemeral identifiers and therefore, they have been used for the EV and CS whose interactions are temporary. The properties of public and peer DIDs are presented in Table 3.1.

DID Type	**Public**	**Peer**
Used By	ER, CSO	EV, CS
Key Source	Ledger	Encoded in DID
Key Resolution	Slow	Fast
Method	did:sov	did:peer

Table 3.1: Comparison of public and peer DIDs

Since ER DID and CSO DID is used to issue credentials that must be verified by third parties, they need to be publicly resolvable. The public DIDs used in the design are based on the Hyperledger Indy [34] because of it provides a full ecosystem for managing DIDs such as ledgers, wallets, encryption standards [26], etc. In Hyperledger Indy, the public DIDs are written in a Hyperledger Indy ledger pool of observer and validator nodes. Hyperledger Indy relies on a permissioned distributed ledger and is managed by trustees, stewards, endorsers, and users. Hyperledger Indy acts as a PKI to reliably resolve DIDs for CSO, ER and DSO by the other entities and thus removes the need for installing device certificates inside EVs and CSs. The public DIDs are denoted using the method did:sov and a 16 byte random string as identifier.

3.1.2 Ring Signatures

Ring signatures and group signatures [13] both allow creation of anonymous signatures. However, group signatures are based on cryptographic algorithms which are considerably more complex than ring signatures. Group signature operations for signature generation and verification take several seconds on mobile devices [43]. Furthermore, group signatures require the participation of all members for group creation and therefore group signatures have not been used in the design.

Ring signatures on the other hand are much simpler operations. Ring signature performance degrades linearly with number of members in the ring. Several kinds of ring signatures schemes providing special properties have been proposed and analysed [38]. However, specialised ring signatures are

generally more complex and a simple ring signatures was sufficient for this thesis.

3.1.3 Credentials

Credentials provided by the Hyperledger Indy are based on ZKP and therefore are significantly resource consuming and unsuitable for deployment on constrained devices. Although they provided features such as selective disclosure and anonymous presentation, these features were not required for the use case. Therefore, the credentials used in the thesis were designed from Linked Data Proofs [19] syntax with detached JWS for signatures. The credentials were designed to be disclosed in full during presentation.

Additionally, presentations in Hyperledger Indy are transferable, which means a presentation can be forwarded by an entity who does not own the credential. This vulnerability can be exploited for a Man-in-the-Middle (MITM) attack. Thus, credentials used in the design expose the holder DID (or set of possible holder DIDs in case of Ring Signature Design). The credentials are presented with a fresh signature created by the secret keys associated with the holder DID mentioned in the credential. Using the same DID for communication which is also included in credentials prove to the other party that the presenter of the credential is the legitimate owner of the credential.

3.1.4 Threat Model

An attacker might try to capture the communication traffic between EV and CS to get information about the EVU such as ER name, amount charged etc. Therefore, all communication except for the connection invitation was encrypted following the principle of least information. The CS_INV_KEY securely bootstraps the connection and the other messages in the protocol are encrypted with keys associated with peer DIDs of EV and CS.

As Bluetooth devices are inexpensive, it is easy for an attacker to install small devices near the station to set up fake CS services to steal credentials and Payment Commitments from the EVs. Another possible attack related to Bluetooth communication could be that the EVU is not plugged in to the correct charging socket and accidentally pays for the charging of the attacker. This seems to be an unlikely attack because the attacker then must be present at the same CS and would be caught. The CS Name and socket number are advertised via Bluetooth and the same are also printed clearly at the CS. Thus, there should be no confusion about which socket the EV is attached to and which CS socket the EV application is communicating with. If there

is a competing CS peripheral advertising the same socket number, the EVU is able to visually inspect if its the correct CS.

Lastly, another threat could be the evasion of full payment by the EV for electricity received from the CS or alternately the CS not supplying the full charge for the payment from the EV. The micropayments scheme described earlier protects both parties from this kind of fraud.

3.2 Assumptions

Assumptions made for the Ring Signature Design are listed in this subsection. The same assumption are also applicable for the Multi-DID Design.

- There is no collusion between the CSO and ER to combine the information stored by them about the EVU to reveal the EVU's identity and other information.
- A distributed ledger is run and managed by stewards which may be entities representing the interests of the parties involved e.g. a consortium of EVU organisation, CSOs, ERs and governmental regulatory authorities.
- The public DIDs for the ER and the CSO are written on the distributed ledger and the associated public keys are rotated periodically to maintain security of the system. The DID values, i.e. ER DID and CSO DID, are known by EVU and CS to be as legitimate.
- Key rotation is performed regularly by the CSO and ER. CSO and ER verification keys are rotated such that already issued credentials do not get invalidated. This is done by coordinating the key rotation time and credential validity.
- The communicating parts of the EV and CS are IoT devices with enough computation capabilities to perform public key cryptography and equipped with low power wireless communication capability such as Bluetooth Low Energy (BLE). They have access to the internet for downloading credentials, although internet access is not required during the charging event.
- The DSO has smart meters installed at each CS providing it with information about total electricity used in a time period.

- CSO issues credentials with correct district ID. If CSO issues credentials with false district ID, CSO would be caught by the DSO when it matches transaction data against smart meter readings for the district.
- A charging station typically has many sockets at a location. However, here a single charging socket is denoted by the abbreviation CS. There maybe many CSs close to each other and owned by the same CSO, but they differ by the given CS Name and socket number, allowing EVUs to differentiate between them. There are also visual indicators that show CSO and ER branding, such that there is no confusion about which charging station serves a particular ER and owned by which CSO.
- There exists secure channel of communication between the CSO, ER and DSO such that they can securely transfer Transaction Logs.

3.3 Credential Generation

In the Ring Signature Design, each month the CSO collects the CS DIDs belonging to all the CSs in the district. Then CSO isses the same CS Credential to each of the CSs. The subject of the credential is a set all the CS DIDs and the District ID. This set of CS DIDs is used by the CSs as the signer set for generating ring signatures. A verifier of the ring signature can be sure that one of the DIDs in the set belongs to the signer but cannot figure out which one. The CS Credential is valid for 1 month. The structure of the CS Credential is detailed in Appendix A. The EV credential generation method is the same as in the Multi-DID Design.

3.4 Exchange Messages and Presentations

The exchange messages used are the same as in Multi-DID Design but the CS Presentation and Payment Commitment have been modified. The CS Presentation is created by generating a ring signature on the EV DID received from the EV. In the Multi-DID design, CS Presentation was discarded after verification but in Ring Signature Design it is used to create the cs-signature field of the Payment Commitment. The cs-signature consists of the EV DID and ring signature obtained from the CS Presentation. In the Payment Commitment of the Ring Signature Design the cs-did field is replaced with the cs-signature field. Other fields in payment commitment are the same as in the Multi-DID Design. The structure of the CS Presentation and Payment Commitment is shown in Appendix A.

3.5 Charging Transaction

The details of the charging transaction in the Ring Signature Design are described here. The steps are indicated in Figure 3.1 and explained as follows:

1. Same as Multi-DID Design in Section 2.7.3.
2. Same as Multi-DID Design.
3. Same as Multi-DID Design.
4. Same as Multi-DID Design.
5. Same as Multi-DID Design.
6. Same as Multi-DID Design.
7. Same as Multi-DID Design.
8. **The CS uses its long lived CS DID and CS Credential from its wallet**. CS also generates a CS Presentation by generating a ring signature on the the EV DID with the other CSs in the district as ring members.
9. Same as Multi-DID Design.
10. EV verifies the CS Credential and the CS Presentation. As the CS Presentation contains only a ring signature, it proves that the presentation creator is one of the CSs in the district without revealing the exact CS. Then, EV generates the Payment Commitment. The hash chain creation procedure is the same as in Multi-DID but **the cs-did field is replaced with cs-signature**. *cs-signature* is the message and signature acquired from the CS presentation in the previous step. The Payment Commitment is then countersigned with private key associated with the EV DID.
11. Same as Multi-DID Design.
12. Same as Multi-DID Design.
13. Same as Multi-DID Design.
14. Same as Multi-DID Design.
15. Same as Multi-DID Design.
16. Same as Multi-DID Design.

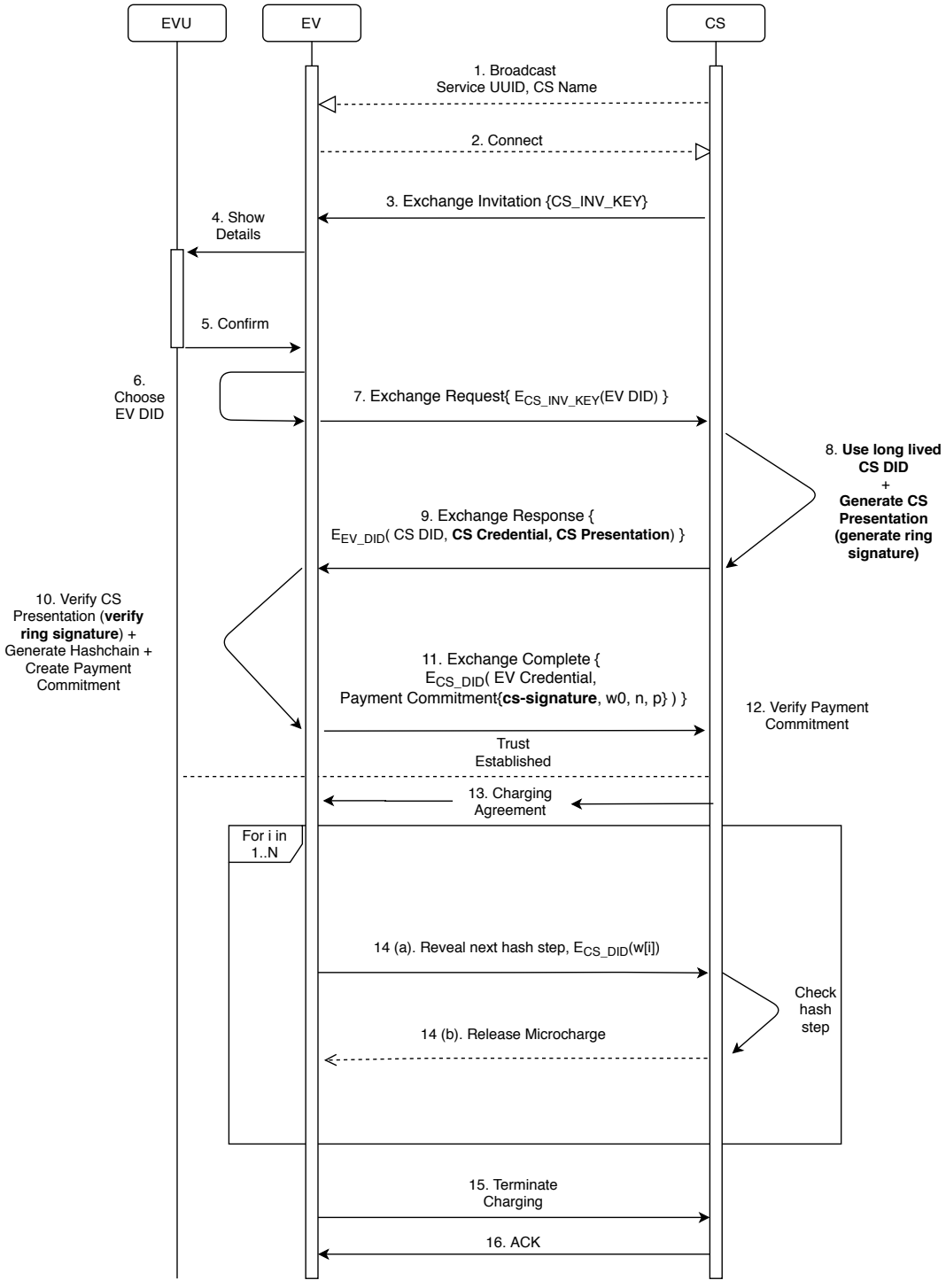

Figure 3.1: New Design using Ring Signature for reducing creation of one-time use DIDs

3.6 Payment Resolution

The collection and delivery of the Transaction Log of charging events is the same as in the Multi-DID Design except that the Payment Commitment and the CS credential are modified slightly as discussed earlier. Although CS and EV know that CS DID is the true signer of the receipt, it is not apparent from the ring signature itself, and the ER and DSO do not learn the CS DID from the ring signature.

1. With the Transaction Log, EV Credential and CS Credential, the CSO is able to:

   (a) Identify the district the transaction happened from the CS Credential. The CSO knows the CS identity as well because it receives the Transaction Log from the CS. However, this does not reveal the EVU's location to CSO as the EV DIDs are different in each transaction and CSO does not know the real identity associated with the EV DIDs.

   (b) Identify the ER of which the EVU is a customer and thus demand payment from the ER for the services rendered.

   (c) Identify the amount of energy used for charging from the amount of charge per step 'p' and the number of micropayments 'k'.

   (d) Identify the time of the transaction from the timestamp in the Payment Commitment.

2. Similarly, with the Transaction Log, EV Credential and CS Credential, the ER is able to:

   (a) Identify only the district the transaction happened from the CS Credential. This is because the ring signature in the CS Presentation does not reveal the CS DID used in the transaction.

   (b) Identify the CSO whose CS provided the charging service and thus make payments to that CSO appropriately.

   (c) Identify the amount of energy used for charging from the amount of charge per step 'p' and the number of micropayments 'k'.

   (d) Identify the EVU who made use of the charging service from the EV DID in the EV Credential and appropriately bill them.

   (e) Identify the time of the transaction from the timestamp in the Payment Commitment.

3. And finally, with the Transaction Log, EV Credential and CS Credential, the DSO is able to:

   (a) Identify only the district the transaction happened from the CS Credential. This is because the ring signature in the CS Presentation does not reveal the CS DID used in the transaction.

   (b) Identify the ER involved in the transaction from the EV Credential and reward the ER appropriately.

   (c) Identify the amount of energy used for charging from the amount of charge per step 'p' and the number of micropayments 'k'. The DSO sums the total energy used per district and decides the degree of the ER's contribution to grid balancing.

   (d) Identify the time of the transaction from the timestamp in the Payment Commitment.

3.7 Limitations

The level of privacy provided by both the Multi-DID and Ring Signature Design depends on the size of the districts and the number of charging stations located in that district. If the district is either geographically too small or has very few CSs located in it, the EVU's location may get compromised.

While in the Multi-DID Design, a new CS DID could be used right after issuing a credential to the CS, in Ring Signature Design, updating a long lived CS DID would require issuing new credential not only to that CS, but to all the CSs in the district. However, updating a single CS DID is an unlikely event as all CS DIDs are updated regularly, and therefore this limitation does not cause any major problems.

A collusion between CSO and ER for monetary gains could be a risk. It has been assumed that there is no collusion between the two but in the real world collusion might happen. The CSO knows the EV DIDs and real identities of CS involved in the charging event and the ER knows the EVUs associated with the EV DIDs. By combining their information the ER and CSO can reveal EVU's location. Another possible threat is that a CS can use non random values for ring signatures and thus undermine the anonymity of the ring signature. However, such ring signatures can be detected by a vigilant EV and the CSO could lose its reputation or face legal action.

Finally, correlation attacks may be possible using other data relating to the use of services. For example, some information may be leaked from network identifiers when the EV acquires the credentials via mobile network

used to access the internet. Other information such as the type and model of the vehicle and the characteristics of the battery system in the EV could also be used to identify the EVU.

Chapter 4

Implementation

This section describes the implementation details for the prototypes of EV and CS developed in the thesis. Firstly, it presents the devices used and their capabilities. Then, it shows the properties of Bluetooth Low Energy specific to Android. Next, it shows architecture details for the two prototypes. Finally, it provides an overview of the software and libraries used for the development.

4.1 Devices

The devices used for the implementation were two Android smartphones — Nokia 8.1 and Nokia 6.1. Both Multi-DID and Ring Signature Designs assume that the EV and CS are capable of performing public key cryptography. Since modern smartphones are quite powerful, they were expected to perform the cryptographic operations easily.

Another advantage of using smartphones is that they have number of communication options. According to the designs, the devices should be able to access the internet to connect to the ledger and acquire credentials from the ER or CS, although the devices do not need internet connection while charging. Furthermore, they should be able to communicate directly with each other through a secure wireless channel as well. Among the options were Near Field Communication (NFC), Bluetooth Low Energy (BLE) and WiFi Direct. Bluetooth Low Energy was chosen as the communication channel as BLE provides low energy consumption and sufficient range. The implementation uses the latest Bluetooth version 5 which supports faster data rates and longer range [17].

The experiments are conducted with the Nokia 6.1 representing EV and Nokia 8.1 representing CS in the charging transaction. Distance between the

phones is 1 metre which is a reasonable assumption of distance between EV and CS. The device information and Bluetooth Low Energy configuration of the phones is shown in Figure 4.1.

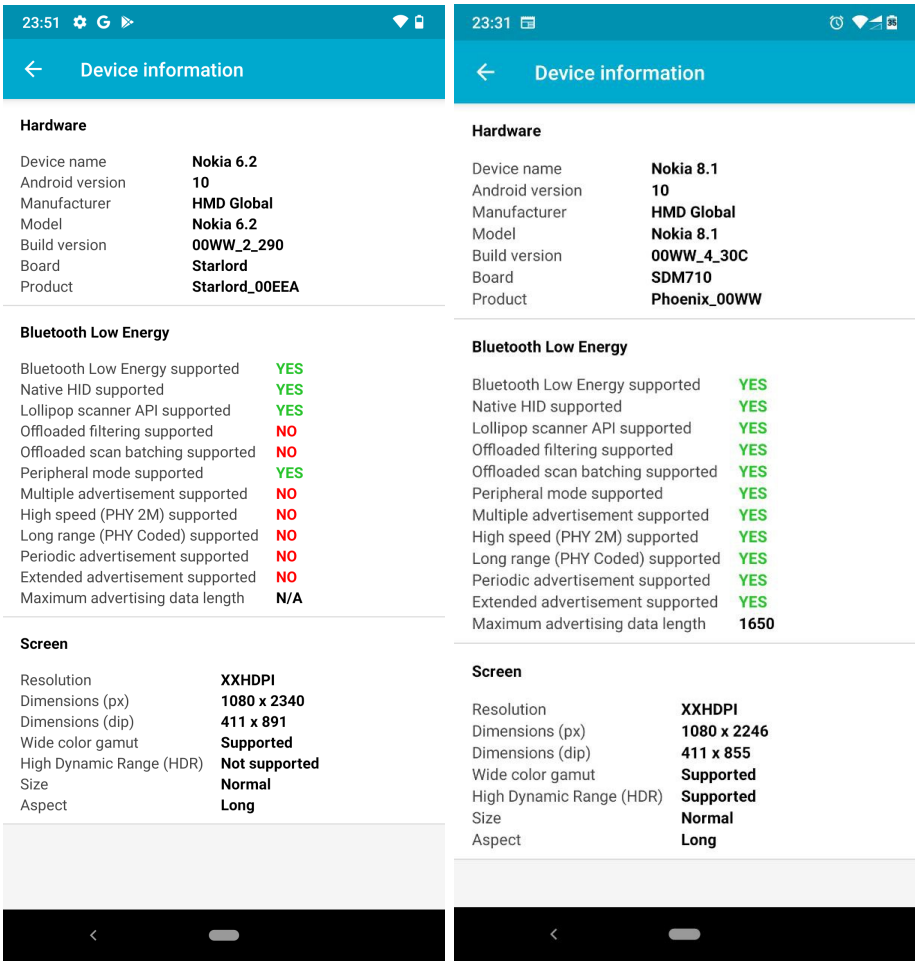

Figure 4.1: Device information and Bluetooth Low Energy configuration for the test phones as shown by the NRF Connect Application [3]

4.2 Bluetooth Low Energy (BLE) in Android

Bluetooth Low Energy version 5 [17] has many features, such as faster physical channel which theoretically doubles the transmission speed compared to BLE version 4, longer range, and error correction codes. However, the actual BLE performance depends on various other connection parameters such as Maximum Transmission Unit (MTU), connection interval, packets sent per

connection interval, write mode, etc. BLE in Android supports a Maximum Transmission Unit (MTU) length of 517 bytes which the maximum length of a single packet. The connection interval for a high priority connection in Android is 7.5 ms and up to 4 packets can sent per connection interval. After connection establishment Android allows negotiation of the connection parameters.

BLE devices mainly have two profiles: 'peripheral' which broadcasts and provides the service, and 'client' which may scan and connect to the peripheral. Data in a BLE peripheral resides in characteristics which are fixed length memory spaces from which data can read and written to. After connecting with a peripheral, the client must perform service discovery which reveals all the characteristics on peripheral. A client may request the peripheral to send update notifications for a specific characteristic [41]. The time to discover a peripheral and perform service discovery depends on the scan mode and pairing state of the devices. For the measurements, the devices are kept in unpaired state to simulate the case when an EV uses a new CS. Pairing state has an impact on transaction time as already paired device use a cached list to discover services and therefore connection establishment is much faster.

Charging stations are connected to the electric vehicles via cables and therefore could use a wired communication protocol. However, this implementation was done with BLE to incorporate future cases such as wireless charging [33]. Another benefit of using BLE is that the charging protocol can begin already as the EV is approaching the CS. BLE on Android however has a limitation that the permission to access location services must be granted to an application using BLE and therefore the user has to trust the application to not misuse location data.

Messages longer than the MTU are required to be in packets in BLE. Furthermore, BLE operations in Android are required to be performed serially as otherwise the newer operation overrides the older operation. Thus, applications using BLE need to implement message buffers for sending long messages and operation queues for serialising operations [59]. The structure of EV and CS buffers and operation queue is shown in Figure 4.2. The implementation uses three characteristics to communicate the messages between the devices: Rx which receives the messages from the EV, Tx which sends information to the EV via notifications and 'stage' which is used to synchronise the progress of the protocol between the 2 devices. In the EV, all BLE operations are first put in operation queue and then serially executed. The write operation divides the message into packets of size equal to the MTU and stores them in the write buffer. The packets are then read from the write buffer and sent serially. For read operations, packets received from the CS

are queued to the read buffer until a message with a end of message indicator is received upon which all packets are combined to recreate the full message. A similar process is followed by the CS as well.

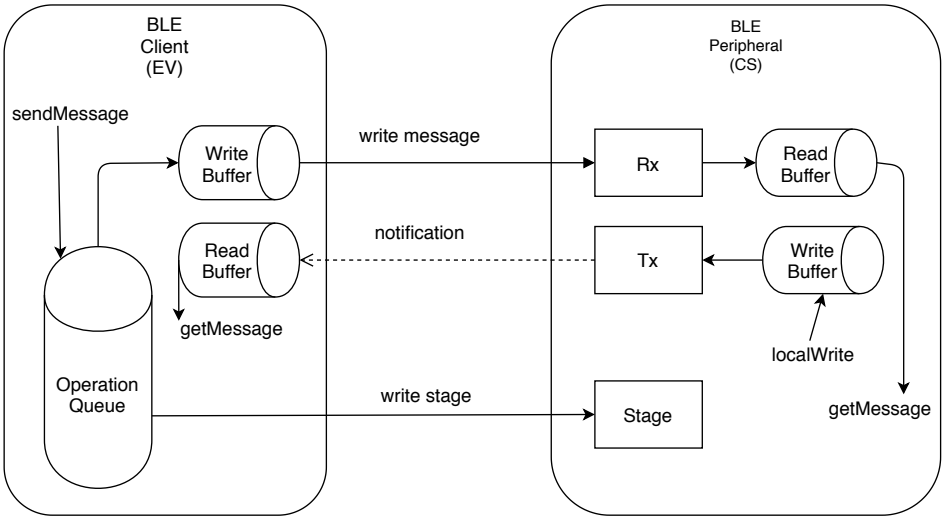

Figure 4.2: Structure of the EV and CS applications showing message buffers, operation queue and characteristics

4.3 System Architecture

The implementation focused on the interaction between the EV and CS during the charging transaction and the rest of the interactions such as EVU registration with ER and CS on-boarding to CSO is assumed to have happened beforehand. The process of acquiring credentials is simulated by generating the credentials on the device. The design of the EV and CS application is described next and their flow diagram is shown in Figure 4.3.

The EV application on being started, initialises the libraries and generates EV credential. Then, the EV scans for nearby BLE services with the target service UUID. Once the target CS is found, it negotiates the BLE connection parameters such as MTU, discovers services and enables notifications on the stage characteristics. Next, the protocol begins which involves exchange of DIDs, credentials and presentations. The EV also performs various cryptographic operations such as signature generation and verification, encryption and decryption, hashing etc. If the mutual authentication succeeds, the EV application proceeds charging with by sending micropayments. At each iteration the EV sends the next hash chain step and waits for the the charge

(which is simulated with a notification update from the CS). The protocol finishes when all hash chain steps have been revealed or if terminated by the user. Similarly, CS also initialises the library and generates CS Credentials on the device. It then broadcasts the service UUID. The charging protocol starts after the discovery and connection creation process as explained before. The CS verifies the EV Credentials and Payment Commitment and on successful verification, the micropayments are received and corresponding notification (representing release a charge) is sent. The code for the implementation is available in its GitHub repository [11] and the user interface of the applications is shown in Figure 4.4.

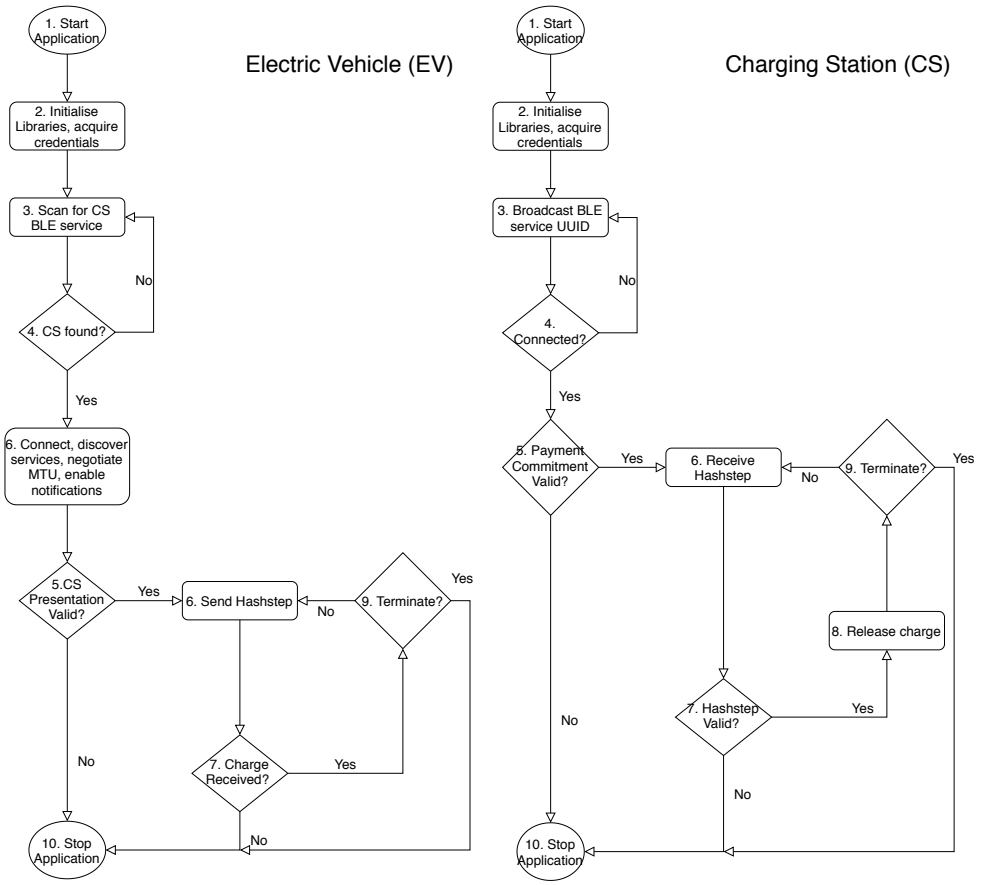

Figure 4.3: Flow diagram of Electric Vehicle (EV) and Charging Station (CS) applications

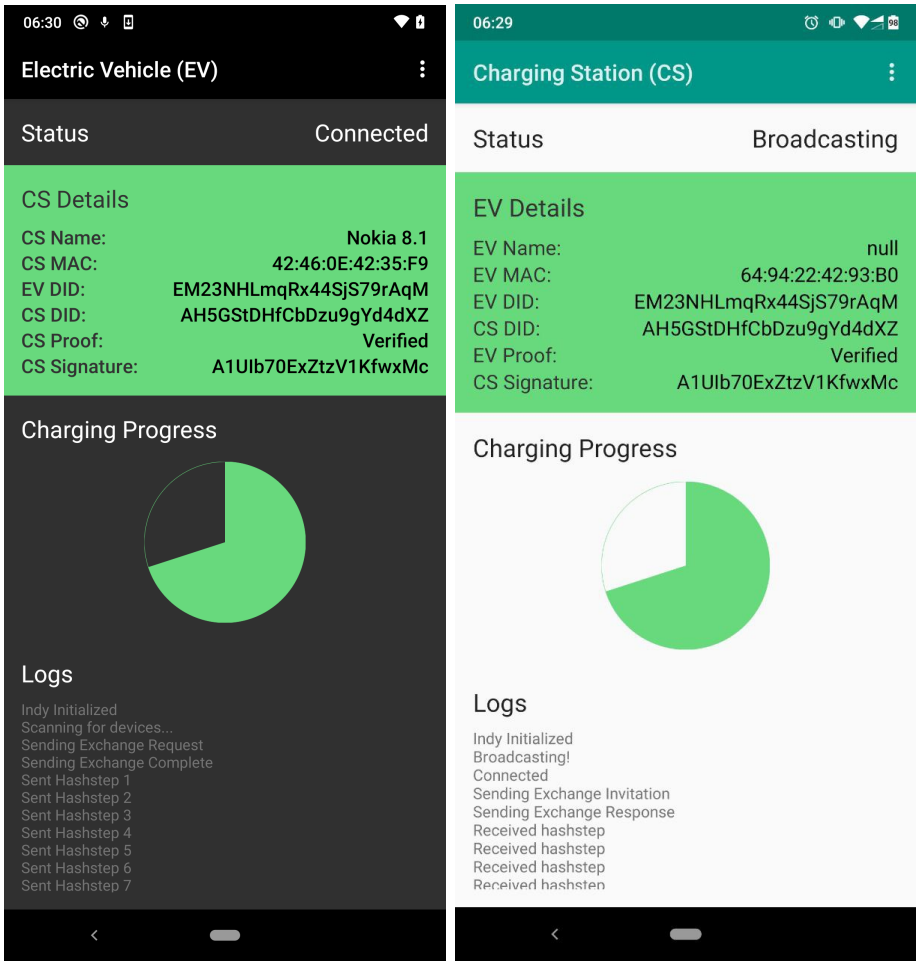

Figure 4.4: User interface of the applications: Electric Vehicle (EV) and Charging Station (CS)

4.4 Software Overview

An overview of the software used such as SDKs, IDEs and libraries have been described here.

4.4.1 Android SDK and Development Environment

Android Studio 4.0.1 was used as the development environment. Android Studio provides various useful features such as visual graphics designer and code completion. Both the EV and CS applications used Android SDK version 28.0.3 for compilation along with Gradle version 3.5.3 for dependency

management.

4.4.2 Kyber Crypto Library

Various cryptographic libraries are available in Android such as Libsodium [23] and Lazysodium [61]. Although these libraries have Ed22519 cryptography functions, none of them had ready to use ring signature implementation. Implementing our own ring signature library would have reduced confidence in the results. Thus, ring signature implementation in the Kyber crypto library [21] which is written in Golang, was used for this thesis.

Kyber crypto library from the Dedis group is managed by programmers from EPFL and Yale. It has been extensively tested from timing and message length attacks. Most importantly it has a ready to use implementation for ring signatures with Ed25519.

As Kyber is written in Golang, cross language bindings were required to integrate into Android. Gomobile [18] was used to generate bindings of Kyber for java. Gomobile generates an Android library file .AAR which can simply be imported to the main application. However, calling cross language functions have some processing overhead which might have increased the timing measurements.

4.4.3 Hyperledger Indy SDK for Android

Hyperledger Indy SDK is available as library C-callable library. In Android, it can used via JNI integration. Hyperledger Indy provide a complete suite of functions for identifier management such as DID creation, wallets, connection to ledgers and cryptographic primitives like encryption, decryption, and signing.

Chapter 5

Results

This section presents the results from the measurements with the prototype implementations. Firstly, it details the test assumptions based on real data from Norway such as the number of EVs, CSs, districts and charging transactions. Secondly, the performance and scalability of ring signatures was compared with regular signatures. Then, the Multi-DID Design and the Ring Signature Design performance are compared on four fronts: identifier creation, credential generation, charging transaction, and billing evidence.

5.1 Test Assumptions

The fleet of electric vehicles in Norway is the largest per capita in the world [48] and their popularity has also led to a highly developed EV charging infrastructure in the country. Thus, Norway was chosen as the model for the test assumptions in this thesis. According to the Norwegian Charging Station Database, NOBIL [39], as of July 2020, Norway has 12,938 regular charging points and 2,539 fast charging points, for a total of 15,477 publicly available charging points. Additionally, at the start of 2020, Norway had 260,692 registered battery electric or plugin hybrid vehicles [57].

According to the survey by the Norwegian EV Association [22], EV owners, who live in smaller houses, often do not have their own garages or designated parking spaces and therefore are more likely to charge their EVs at public CSs. The highest percentage of public charging station users, 28%, are users living in apartment buildings—they claim to charge their EVs daily or weekly at public charging stations.

Based on the survey data, and taking a conservative approximation, the calculations in this thesis are done under the assumption that all the EVs in the country are being used regularly and 28% of these EVs are charged

daily at public CSs. Thus, assuming charging station usage is spread evenly throughout the country, each CS was calculated to performs 5 charging transactions per day. According to Statistics Norway [64], Norway has 356 municipalities and the test environment was assumed to have the same number of energy districts, although in reality these districts are created by the DSO and may be different from municipal districts. Although the number of charging stations is greater in the southern region of Norway, it was assumed that there are the same number of charging stations in each district. The number of CSs per district was calculated to be 44 and rounded up to 50 for the calculations. Therefore, the ring size of the set created by the CSs in each district is also 50. The assumptions have been summarised in Table 5.1.

Entity	Count
Total EVs	260 692
Total CSs	15 477
Number of districts	356
CSs per district	50
Ring Size	50
Charging transactions (per CS per day)	5

Table 5.1: Number of EVs, CSs, districts and transactions used in calculations in this thesis

5.2 Ring Signatures

A key factor affecting the performance of ring signatures is the ring size, which is defined as the number of members in the anonymity set of the ring signature. As the ring size increases, more resources are required for signature generation as the signature value must be valid for each public key in the anonymity set and similarly, signature verification also requires more resources as the signature must be verified against each of the public keys in the anonymity set. The signature length also increases with ring size because it contains the contribution value for each of the signers in the anonymity set.

A type of Edwards-curve Digital Signature Algorithm (EdDSA), Ed25519, which combines the Curve25519 and SHA-512 hashing algorithm, was used for regular signatures in this thesis. The implementation is provided by the Kyber crypto library. A signature with this algorithm is 64 bytes long and

it takes 0.875 ms for generation and 1.06 ms for verification of signature on 32 kilobytes of data on the Nokia 8.1 test device. The results are in agreement with expectation as it is known that verification is a bit slower than generation in EdDSA [1].

The measurement for ring signatures was also done using the same EdDSA algorithm implemented in the Kyber crypto library although it uses a slightly faster Blake 2xb [28] hashing algorithm. However, this difference should have negligible effect on the results as the hashing operation (SHA-512) is more than 100 times faster than signing and more than 300 times faster than verification with Ed25519 [27]. A measurement of ring signature performance with respect to changing ring size was executed on the same device (Nokia 8.1) and the results are shown in Table 5.2. The measurements start from ring size of 1 for completeness, but such a ring size is unlikely to have any application. The length of the ring signature increases linearly with growing ring size throughout the range. The signature length for the ring size of 50 is about 1.6 kilobytes (KB). However, ring signature length of ring size larger than 200 becomes quite large. At ring size of 1,000 members, the signature length grows to 32 KB which is the same size as the data signed.

Ring Size	1	2	5	10	20	50	100	200	500	1 000	2 000	5 000	10 000
Signing (ms)	0.423	1.26	3.09	6.39	13.3	33.3	66.9	134	335	670	1 340	3 410	7 140
Verification (ms)	0.785	2.01	3.46	6.76	13.3	33.1	66.4	132	331	659	1 320	3 366	6 670
Signature Length (bytes)	64	96	192	352	672	1 632	3 232	6 432	16 K	32 K	64 K	160 K	320 K

Table 5.2: Effect of ring size on signature generation time, verification time and signature length.

Based on the Table 5.2, the relationship of ring signature generation and verification times with ring size is shown in Figure 5.1. The figure shows that the signature generation time and signature verification time increase linearly with increasing ring size for the majority of the range. The signature verification time is greater than generation time for ring sizes smaller than 20, but for larger ring sizes the verification time decreases slightly and becomes almost the same as the signature generation time. This is in contrast with regular elliptic curves signatures, which usually have verification time larger than signature generation time [1].

On the whole, the signature generation and verification times are roughly the same as both involve almost the same mathematical operations except for one inversion operation on the signer's trapdoor permutation during signature generation. Even for very large rings of size 10,000 members, the time for ring signature generation and verification keeps growing linearly. However, the time grows to several seconds which would significantly degrade the user experience. For ring size of 50 members, which is also the ring size of

CSs in a district in the thesis, a signature can be generated or verified in about 34 ms. Thus, having negligible effect on the user experience.

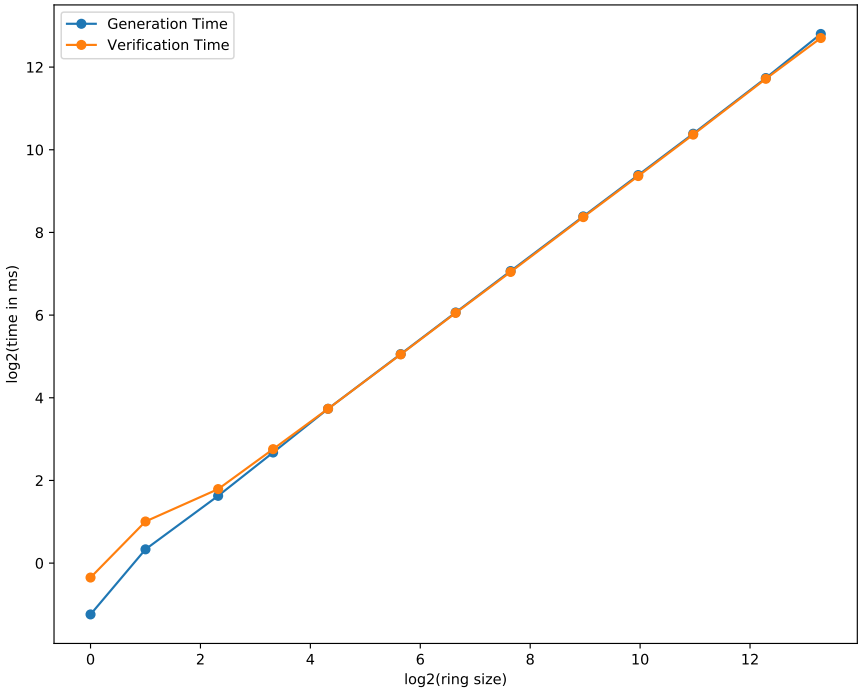

Figure 5.1: Ring Signature generation and verification time with respect to ring size

The findings show that ring signatures are scalable operations and easily support large ring sizes of more than 1,000 keys. Ring Signatures are computationally inexpensive operations as signing and verification operations for a ring size of even 1,000 members can be done in well under a second. However, storage and transmission of ring signatures might become challenging for bigger ring size as signature length increases and consume significant memory resources. Thus, ring signatures with moderate ring sizes of up to 100 keys are suitable for use on modern mobile devices.

5.3 Identifiers

Two different types of DIDs are used in the both the Multi-DID and Ring Signature Designs — Public DIDs and Peer DIDs. The generation time and storage requirement for the peer and public DIDs are the same as both are derived from a public-private key pair. An Ed25519 key pair consists of a 32 byte public key and a 32 byte private key, thus making a total storage requirement of 64 bytes [52]. The average time require to create a peer DID is 0.287 ms on the Nokia 8.1 device. The major component of the DID creation is the creation of the elliptic curve key pair. Creation of public DID also involves writing the associated keys in the ledger but they can be written before the protocol starts and therefore ledger operations do not affect the EV charging performance.

Since the public DIDs are pre-generated and the peer DID usage by EV is same in both designs, only peer DID usage by the CS was measured. Table 5.3 shows the comparison of computation performance on the Nokia 8.1 device and the memory required for CS DID in the Multi-DID and the Ring Signature Design.

Conditions	**Multi-DID**	**Ring Signature**
Period: 1 Day Districts: 1 CSs: 1 Transactions: 5	DIDs: 5 Time: 1.43 ms Size: 320 B	DIDs: 1 Time: 0.287 ms Size: 64 B
Period: 1 Month Districts: 1 CSs: 50 Transactions: 7500	DIDs: 7 500 Time: 2.1 s Size: 480 KB	DIDs: 50 Time: 14 ms Size: 3.2 KB
Period: 1 Month Districts: 356 CSs: 15 477 Transactions: 2 321 550	DIDs: 2 321 550 Time: 11 min Size: 150 MB	DIDs: 15 477 Time: 4.4 s Size: 0.99 MB

Table 5.3: Comparison of peer DID usage in CSs — per CS per day, in a district per month, and for the whole country per month. The number of DIDs generated, their generation time and the size of storage required are shown for each condition.

Firstly, the peer DID usage by a single CS in a typical day is presented. Based on the assumptions, A CS performs, on average, 5 transactions. In the Multi-DID Design, 5 DIDs are created per day and need to be stored on

the CS device. On the other hand, Ring Signature Design requires creation of only 1 DID. The storage and computation requirements are low in both designs.

Next, the number of DIDs created and associated performance in the time period of a month in a district with 50 CSs is shown. In the course of a month, 7,500 charging transaction take place at different CSs in a district. In the Multi-DID design, the 7,500 DIDs need to be generated compared to just 50 in the Ring Signature Design. These DIDs need to be stored by ER, DSO and CSO as part of payment commitment in the Multi-DID design.

Finally, DID requirements for all districts in the country in the course of a month are shown. In 356 districts, 2,321,550 transaction occur and the same number of DIDs are generated in the Multi-DID Design. In contrast, Ring Signature Design requires creation of just 15,477 DIDs which require 150 times less storage than the other design. Still, the storage requirements are low as well as the DID generation time for both the designs. A CSO with the computing power of a server could perform these operations even faster. Thus, peer DIDs are suitable for use even in large numbers.

5.4 Credentials

Two types of credentials are used in the Multi-DID and Ring Signature Design — EV credential and CS credential. The properties, creation time and storage requirement of these credentials have been summarised in Table 5.4.

Property	**EV Credential (Same in both designs)**	**CS Credential (Multi-DID Design)**	**CS Credential, Ring Size = 50 (Ring Signature Design)**
Attribute	1 EV DID	1 CS DID, District ID	Set of 50 CS DIDs, District ID
Validity	5 days	3 days	1 month
Usage	1 transaction	1 transaction	multiple transactions
Issuer	ER	CSO	CSO
Creation Time (ms)	5	6	6
Size (bytes)	651	667	2987

Table 5.4: Overview of credentials issued to EV and CS before the charging transaction for proving their authorisation to each other.

The EV credential is the same in both designs. The EV credential is issued by the ER and is used to authorise the EV DID and thus allows the EV using it to charge at specific CSs. It is acquired by the EV before the transaction and is valid for 5 days. As EV DID is used just once per transaction, the corresponding EV credential is also used for only 1 transaction.

The CS credential on the other hand, is different in the two designs. In the Multi-DID design, the CS credential authorises the CS DID and also

asserts the district ID of the CS. Once issued by the CSO, it is valid for 3 days but can be used only once. In the Ring Signature Design, CS credential specifies a set of 50 CS DIDs belonging to the district along with the District ID. This credential, in contrast, is valid for a month and can be reused.

On the Nokia 8.1 test device, EV credential took 5 ms to create while taking 651 bytes to store. The creation of CS credential took 6 ms for both the Multi-DID and Ring Signature although the size was different 667 bytes and 2,987 bytes respectively. The creation time is the same despite different size because the major time consuming process is signing which is dependant on the fixed sized hash of the data to be signed.

Applying the above properties of credentials to the test assumptions, the credential generation time and size requirement was determined for 1 CS per day, for 1 district over a month and for all districts over a month as shown in Table 5.5.

Conditions	**Multi-DID**	**Ring Signature**
Period: 1 Day Districts: 1 CSs: 1 Transactions: 5	Credentials: 5 Time: 30 ms Size: 3.3 KB	Credentials: 1 Time: 6 ms Size: 2.9 KB
Period: 1 Month Districts: 1 CSs: 50 Transactions: 7500	Credentials: 7 500 Time: 45 s Size: 5 MB	Credentials: 1 Time: 6 ms Size: 3 KB
Period: 1 Month Districts: 356 CSs: 15 477 Transactions: 2 321 550	Credentials: 2 321 550 Time: 4 hrs Size: 1.5 GB	Credentials: 356 Time: 2.1 s Size: 1 MB

Table 5.5: CS Credentials generation per CS per day, per district per month and all districts per month. The number of credentials generated, their generation time and the size of storage required are shown for each condition.

Firstly, in the Multi-DID design, a single CS uses 5 credentials during a typical day whereas in the Ring Signature Design, since credentials are reused, only 1 credential is used. The credentials are stored on the CS device and require around 3 KB of storage space in both designs. Secondly, the number of credentials issued to CSs per district in the Multi-DID Design increases to 7,500 when calculated over the course of a month while in the Ring Signature Design, the number of credentials stay the same. Finally, when comparing for all districts over a month, the difference is further multiplied

with Multi-DID design using 1.5 GB of storage and 4 hours of computation time compared to just 1 MB of storage and 2.1 seconds of computation time by the Ring Signature Design.

For the billing purposes, the CS forwards the Transaction Logs to the CSO at end of the month. The CSO stores all the CS credentials it has issued and also sends them to the ER and DSO at the end of each month. Thus, CSO, DSO and ER require similar storage space to store all CS credentials used in the past month. In the Multi-DID design, 1.5 GB of storage is required by each of them, while in Ring Signature Design, just 1 MB of storage is required.

Thus, Ring Signature Design leads to a massive saving in credentials storage and computation requirement. It leads to 7000 times less computation time and 1500 times less storage requirement. Although the requirements for the Multi-DID can be easily met with modern computing hardware, Ring Signature Design is still far more efficient.

5.5 Charging Transaction

This section presents the performance measurements of the charging transaction and compares the Multi-DID and Ring Signature Design. First, the test methods and conditions have been defined. Next, a performance measurement of the charging transaction from the perspective of the EV is presented by calculating both the time spent in Bluetooth transmission and in cryptographic operations. Finally, a detailed measurement of time spent per cryptographic operations has been shown to identify the most expensive operations.

The prototype implementations of EV and CS was run on the Nokia 6.1 device and Nokia 8.1 device respectively. Measurements were taken for the Bluetooth connection establishment, time spent in transmission of messages and the time spent in computation as shown in 5.6. The respective DIDs and credentials were already generated before the transaction. The CS is activated first so that it advertises its BLE peripheral service. The message transmission and reception time was calculated from the EV while the computation time was calculated at both the EV and CS. The test was repeated 5 times and then average values were taken. The EV and CS devices were set 1 metre apart.

The Bluetooth connection establishment took 1,814 ms with majority of the time spent in connection established after finding the CS and while discovering services. The Bluetooth connection establishment makes the bulk of the whole transaction time.

Event / Message	Size (bytes)	Transfer time (ms)	Computation time (ms)	Combined time (ms)
Connection establishment				**1814**
Find CS BLE peripheral				128
Establish connection				874
Discover services				764
Enable notifcations				33
Request MTU				15
Charging protocol (Multi-DID)	**3465**	**225**	**162**	**386**
Receive Exchange Invitation	166	82	3	85
Send Exchange Request	254	2	43	45
Receive Exchange Response	1535	129	52	181
Write Exchange Complete	1510	12	63	75
Charging protocol (Ring Signature)	**10855**	**263**	**330**	**593**
Receive Exchange Invitation	166	89	5	94
Send Exchange Request	254	2	46	48
Receive Exchange Response	6749	146	222	368
Write Exchange Complete	3686	26	57	83
Total (Connection + Multi-DID)				**2200**
Total (Connection + Ring Signature)				**2407**

Table 5.6: Comparison of EV charging event for the Multi-DID and Ring Signature Design from the perspective of EV

Compared to the Multi-DID implementation, the Ring Signature implementation transferred 3 times as much data. The time spent in data transfer was however only slightly more as the BLE throughput increases for longer messages. About double the amount of time was spent doing cryptographic operations such as signing presentations and encrypting messages. Overall, the Ring Signature implementation took 9.4% (207 ms) more time for the whole transaction. Thus in practice, Ring Signature implementation is as fast as the Multi-DID implementation.

Figure 5.2 shows the operations performed and the time taken for each message in the charging protocol in the Multi-DID and Ring Signature Design. The labels — Req, Resp and Cmt, refer to the messages Exchange Request (introduced in Step 7 of Section 2.7.2), Exchange Response (Step 9) and Exchange Complete (Step 11) respectively. The value for hash operation in the figure shows only the hashing operations performed as part of the micro transaction setup. Other operations such as signature generation and verification internally perform hashing operations as well but they have not been counted separately. Overall, the time spent in cryptography is larger in the Ring Signature Design for both the EV and the CS with the major increase due to ring signature generation in CS and its verification by the EV. The addition of ring signatures increases the size of the messages but it does not significantly increase the encryption and decryption times and they stay roughly the same.

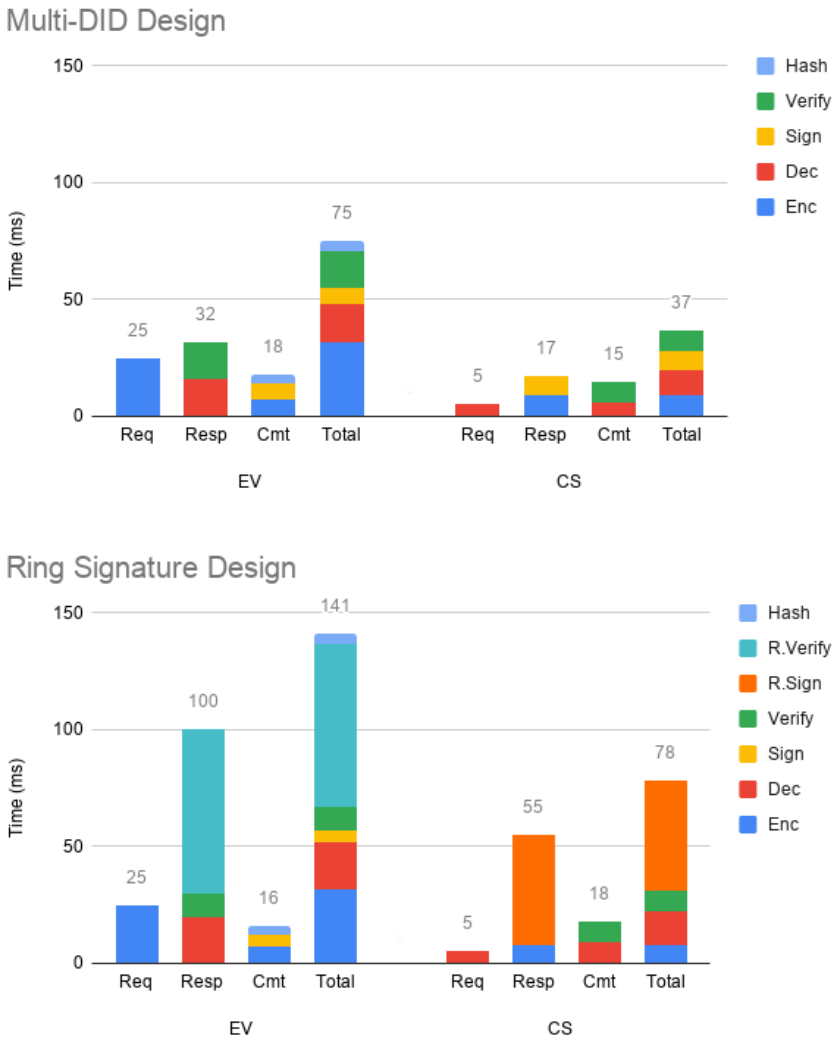

Figure 5.2: Time taken by cryptographic operations during charging transaction in the Multi-DID and Ring Signature Design

5.6 Payment Resolution

This section presents the difference between the storage and processing requirements of the CSO, ER and DSO for identifying and verifying the entities related to a payment relationship for both the Multi-DID and Ring Signature Design. Firstly, the Transaction Log is defined which is the package of information sent along with credentials to identify and authenticate transaction events. Then, the storage requirements per transactions are shown and

requirements for all transactions are interpolated. Finally, the processing required to verify a single transaction is presented and total requirement for all transactions is calculated.

The overview of Transaction Logs in the Multi-DID and Ring Signature Design is presented in Table 5.7. Transaction Log consists of payment commitment, last hash chain step and step number. The major cause of the difference in size of Transaction Logs in the two designs is the large size of ring signature created by the CS included in the payment commitment. This makes a Transaction Log of the Ring Signature Design more than 5 times as large as a Multi-DID Transaction Log.

Entity	Size (Multi-DID)	Size (Ring Sign)
Payment commitment	438	2627
Last hash chain	44	44
Step number	2	2
Total	484	2673

Table 5.7: Details of Transaction Logs and their size (bytes)

The Table 5.8 shows the collection of billing evidence stored by the CSO, ER and DSO for payment resolution in the 2 designs per transaction. In the table, the storage requirement is shown for 1 Transaction Log and associated credentials.

Although only one CS credential is enough to support all the transactions for the month in a district, individual Transaction Logs are much larger in the Ring Signature Design as compared to Multi-DID design, and the total space required is larger. Supposing n is the number of transactions and d is the number of districts, storage required by the CSO for the Transaction Log and credentials can be calculated. In case of Multi-DID,

$$storage = (Log + EVCred + CSCred) \cdot n$$

or

$$storage = (1802) \cdot n$$

On the other hand, in Ring Signature Design,

$$storage = (Log + EVCred) \cdot n + (CSCred) \cdot d$$

or

$$storage = (3324) \cdot n + (2987) \cdot d$$

From the above 2 equations, it can be deduced that storage requirement for Ring Signature Design is larger for any number of transactions. For instance, in a district with 50 CSs and 7,500 transactions, Multi-DID Design has a storage requirement of 13.5 MB whereas Ring Signature has a storage requirement of 25 MB. For all districts with 356 districts and 2,321,550 transactions, the storage requirement is 4.2 GB and 7.7 GB respectively. For very large number of transactions, Ring Signature storage requirement is 1.8 times that of Multi-DID Design.

Multi-DID Design					
Verifier	Received	Size (bytes)	Has	Size (bytes)	Total (bytes)
CSO	Log + EV Cred	1135	CS Cred	667	1802
ER	Log + CS Cred	1151	EV Cred	651	1802
DSO	Log + EV Cred + CS Cred	1802	-	0	1802
Ring Signature Design					
Verifier	Received	Size (bytes)	Has	Size (bytes)	Total (bytes)
CSO	Log + EV Cred	3324	CS Cred	2987	6311
ER	Log + CS Cred	5660	EV Cred	651	6311
DSO	Log + EV Cred + CS Cred	6311	-	0	6311

Table 5.8: Billing evidence collection and storage by CSO, ER, and DSO for payment resolution per transaction

The Table 5.9 shows the cryptographic operations performed by the CSO, ER and DSO for payment resolution in the two designs per transaction. The verification time of a Ring Signature Transaction Log is much longer than a Multi-DID due to the large time take by the ring signature verification.

Taking a similar approach as before with storage requirement, time required to verify n transactions in d districts by the CSO can be calculated as :

In case of Multi-DID,

$$time = (EVCred + EVSign) \cdot n$$

or

$$time = (9) \cdot n$$

On the other hand, in Ring Signature Design,

$$time = (EVCred + EVSign + CSRingSign) \cdot n$$

or

$$time = (82) \cdot n$$

In other words, Ring Signature Transaction Log verification takes 9 times as much time as in Multi-DID Design. For all districts with 2,321,550 transactions it takes 5 hrs 48 min for Multi-DID whereas it takes 52 hrs 54 min.

Multi-DID Design					
Verifier	EV Cred	EV Sign	CS Cred		Total Time (ms)
CSO	7	2	-		9
ER	-	2	10		12
DSO	7	2	10		19
Ring Signature Design					
Verifier	EV Cred	EV Sign	CS Cred	CS R. Sign	Total Time (ms)
CSO	7	5	-	70	82
ER	-	5	10	70	85
DSO	7	5	10	70	92

Table 5.9: Computation time (ms) for verification of the EV Credential, CS Credential, EV's signatures on Payment Commitment, and CS's ring signatures on CS Presentation as performed by CSO, ER, and DSO while performing payment resolution

Therefore, although quantity of CS credentials are reduced in the Ring Signature Design, the total storage requirement is almost double the baseline requirement. This is mainly due to large size of the ring signature. Since smaller ring size leads to smaller signatures, smaller districts should be preferred. The performance degradation in transaction verification is worse — taking 9 times as much time.

Chapter 6

Discussion

This section analyses the results from the previous section and answers the research questions of this thesis.

RQ1: ***How should the charging events be logged and signed with Ring Signatures without revealing personal information of the user and yet prove the authenticity of the charging transaction to the CSO, ER, and DSO?***

Each charging event leads to the creation of a Transaction Log. The Transaction Log consists of a Payment Commitment, last hash chain step and the step number as discussed in Section 3.6. The Payment Commitment is signed by the EV using an EV DID which is unique per transaction. The Payment Commitment also contains the ring signature by the CS using its long lived CS DID. Additionally, the CS collects the EV Credential sent by the EV to authenticate the EV DID.

The Transaction Log is shared with the CSO, ER, and DSO, and they verify the logs according to their purposes as detailed in 3.6. Firstly, the CS sends the Transaction Logs to the CSO. The CSO needs to identify which ER is associated with the transaction. The CSO verifies the signature on the Payment Commitment by the EV and the EV Credential which proves that EV DID used was authorised by the ER and the owner of the EV DID has created the Payment Commitment. Since the EV DID and EV Credential are different for every transaction, the CSO cannot correlate the EV identity over time and thus, the EVU's location privacy remains protected. The CSO also verifies the ring signature in the Payment Commitment which proves that the charging event happened at a valid CSs using the CS Credential which earlier had been issued to CSs and is also kept stored by the CSO. Thus, the CSO can reliably claim payment from the ER for providing the charging service without knowing the identity of the EVU.

At the end of the month, the CSO sends the Transaction Logs along with the CS credentials to the ER. The ER verifies the ring signature in the Payment Commitment of each transaction and the CS credential for each district. If the ring signature is valid for the anonymity set provided by the CS credential then it proves that the charging event happened at a CS owned by the CSO and in the district mentioned in the CS credential. Verification of the signature by the EV DID on the Payment Commitment proves that a valid customer of the ER authorised the transaction. The ER is able to bill the correct EVU as the real identity associated with an EV DID is known to the ER. However, the ER cannot figure out the exact CS involved in the transaction from neither the ring signature nor the CS credential. Therefore ER is unaware of the exact location of the EVU at the time of charging.

Finally, the ER sends the Transaction Log along with both the EV and CS Credentials to the DSO. Following a similar process, the DSO verifies the signature made by the EV on the Payment Commitment and the EV Credential proving that an authorised ER customer charged its EV. The DSO then verifies the ring signature and the CS Credential, which proves that the charging event took place at a valid CS and in the district mentioned in the CS Credential. With this information, the DSO is able to access the extent of ER's contribution in grid balancing per district and rewards it accordingly. However, the DSO neither knows the real identity of EVUs associated with the EV DID nor does it know the identity of the CS involved in the transaction from the ring signature. Hence, all three service providing entities, the CSO, ER, and DSO are assured that the transactions are valid and happened at specified districts without knowing the exact location of the EVUs.

RQ2: ***How does the use of Ring Signatures along with Decentralized Identifiers, as opposed to using just Decentralized Identifiers, affect the protection of privacy for EVUs in the use case?***

In the Multi-DID design, the CS DID is present in the Transaction Logs and therefore, it is known by the CSO, ER and DSO as shown in Section 2.7.4. As the CS DIDs are unique per transaction, the ER and the DSO cannot correlate CS identity. On the other hand, in Ring Signature Design, a ring signature is present in the Payment Commitment and the ER and DSO do not know the CS DID used for the transaction. Although, the CSO may infer the CS DID used from by checking the source of the Transaction Log, it cannot prove this to another party. The location privacy of the EVU depends on the fact that the identity of the CS used for charging is kept hidden. In both designs, the CSO knows the relationship between the CS DID and the

location of the CS using it.

One clear difference, therefore, in the Ring Signature Design, is that the CSO does not know the exact CS involved in the transaction. In the Multi-DID design, as the identity and location of a CS using particular CS DID is known to the CSO, it could reveal this information along with the associated EV DID to others, thus exposing EVU's location at the time of the charging event. Therefore, the Ring Signature Design is slightly more privacy protecting for the EVU.

In contrast, the CSO, ER and DSO in both designs, know the EV DID from the Transaction Log. The CSO and DSO cannot correlate EVU's identity as the EV DIDs are unique per transaction. Only the ER can identify transaction belonging to a EVU as it knows the EVU's identity associated with each EV DID.

However, the ER cannot figure out the exact EVU location from the Transaction Logs in either of the designs. In the Multi-DID Design, the CS DID is used only once per transaction which made correlation impossible if random DIDs are chosen by the CS. In the Ring Signature Design, the ring signature along with the CS credential proves that the CS involved in the transaction is one of the CSs in the district. Correlation from the ring signature itself is hard as each instance of the signature is generated on a fresh message. Thus, in both designs, the EVU's location is known only to the accuracy of the district and the same level of privacy is provided to the EVU.

RQ3: *What is the effect of using Ring Signatures on the resource consumption and the transaction time of the system in the use case?*

Ring Signatures, on their own, are more resource intensive than simple signatures. The ring signatures resource consumption varies with the number of possible signers in the anonymity set. According to Section 5.2, for a ring size of 50, a ring signature is about 35 times slower for generation and verification, and the size of the ring signature is about 250 times larger (1.6 kilobytes) compared regular signatures (64 bytes).

As shown in Section 5.3, in the Ring Signature Design, fewer CS DIDs are required to be generated and stored. Only one CS DID is required per CS per month as compared to new CS DIDs for each transaction in the Multi-DID Design. Thus, for all districts, 150 times less storage and processing resources are required. Fewer number of CS DIDs also lead to creation of fewer number of credentials by the CSO. Although in Ring Signature Design CS credential is much larger (2988 B), far less number of credential are

required. Monthly storage requirement for all credentials in the 356 districts is about 1 MB compared to 1.5 GB for the Multi-DID Design. Therefore, when comparing resource consumption for just the DIDs and credentials, Ring Signature Design performs better.

However, for the Transaction Logs, the situation is different. The Ring Signature Design creates much larger Transaction Logs, primarily due to the bigger size of the ring signature. From Section 5.4 and Section 5.6, in the Ring Signature Design, the CSO requires 3.5 GB of more storage space for the Transaction Logs created in a month for all districts. Subtracting the storage improvement for the credential storage (1.5 GB) for the same period, the overall storage requirement in Ring Signature Design is 2 GB (total 7.7 GB) more than the Multi-DID design.

The payments resolution which involves verification of the signatures in the presentation and credentials is much slower in the Ring Signature Design. As seen in Section 5.6, charging transaction verification for all district in a period of one month requires 53 hours compared to 6 hours on the test hardware. Although this computation is not too challenging to compute for a server, the Ring Signature Design does perform significantly worse, taking 9 times the computation power as the Multi-DID Design.

The total transaction time of charging event for the Ring Signature Design is 2,407 ms compared to 2,200ms for the Multi-DID Design. Out of the total, 1,814 ms is used in establishing the Bluetooth connection and the rest of the 593 ms for the creation of presentations and exchanging credentials across EV and CS. Since the time difference is between the designs is just 207 ms (9.4%), it does not significantly affect the user experience.

RQ4: ***What effect do Ring Signatures have on the deployability of the system on constrained devices?***

The main factors affecting the deployability of a system on constrained devices are computation, power consumption, storage, and communication. Public key cryptography is resource consuming and not all constrained devices are able to perform them. Therefore low computation requirement is extremely important. As seen in Section 5.2 and Section 5.3, ring signature consume more resource in both signature generation and their storage than creating new DIDs for each transaction. Ring signatures also require more computation for verification as compared to regular signatures. As seen in Section 5.5, without the Bluetooth overhead, the computation time is significantly larger with ring signatures while performing a single transaction on sufficiently capable devices.

Power consumption may become a concern as ring signatures are rela-

tively more complex operations. As seen in Section 5.6, about 70 ms is spent in a single ring signature verification as opposed to just 5 ms for regular signatures. Thus, power constrained devices may exhaust way faster while using ring signatures.

Using the equations in Section 5.6, the on-device storage for a CS in the Ring Signature Design can be calculated to be 19,607 bytes compared to 9,010 bytes for the Multi-DID Design. Thus, constrained devices implementing ring signatures need to have bigger memory space. For processing bulk transactions, the servers supporting the constrained devices also require about double the storage space.

More data needs to be communicated when using ring signatures as the larger size of the signature leads to longer messages. As seen in Section 5.5, the transmission of longer messages have little effect on the total transaction time because the encryption and decryption operations and the Bluetooth transfer speeds are relatively quite fast.

Therefore, ring signatures may be deployed on constrained devices when certain conditions are met, such as sufficient power availability, enough hardware capability to perform public key cryptography, and larger storage capacity. If the devices are resource constrained, then the Multi-DID Design is most likely a better option.

RQ5: ***How do Ring Signatures affect the use of Decentralized Identifiers on a broader scale?***

As discussed in Research Question 2, ring signatures provide slightly better privacy but it does end up consuming significantly more resources. In the Ring Signature Design, the overall performance for a single transaction does not degrade much but for bulk operations, such as transaction verification, the difference is substantial. As a very similar level of privacy is provided by the Multi-DID Design for far less resource consumption, using just DIDs is suitable for most cases.

However, using ring signatures might be worth the extra effort in certain situations. Ring signatures might be used in situations which have risks involved of an entity revealing the real identity behind a DID (such as a dishonest CSO). Alternatively, it could be the situation where DIDs are long term identifiers of the entity and ring signatures facilitate the creation of untraceable Transaction Logs without the need to change the DIDs frequently. Thus, its applicability goes beyond situations such as fixed location services. Ring signatures can be used by not only the services but users as well e.g. they can be used in systems to implement anonymous group access of a shared resource.

As seen in the Ring Signature Design, the implementation of ring signature into existing systems using DIDs is easy and requires minimal changes. However, in all applications the ring sizes must be kept small (below 100) to keep the resource consumption low. Of course, smaller ring sizes also means less privacy and therefore, ring signatures use appears to be limited on a broader scale. Ultimately, the choice of design to use depends on the privacy requirements of the system and the resource limitations of the devices and a cost-benefit analysis is advised before making the decision.

Chapter 7

Future Work

The future work includes writing a research paper [6] based on the results of the thesis and compare the Ring Signature Design with the Multi-DID Design in even more detail. The libraries used in the current implementation, Hyperledger Indy and Kyber, had some cross-language overheads and therefore, the prototypes will be re-implemented with native Java functions to get both better performance and more accurate measurements for the designs.

Apart from the paper, ring signatures will be examined as a more general method for anonymous authentication, for example by finding more use cases where ring signatures are better suited. This thesis used simple ring signatures so another direction to continue would be to use more advanced kinds of ring signatures such as Linkable Ring Signatures. In the EV charging use case, linkable ring signatures for the EVs can be used. It is of interest to see if ring signature performance can be improved, eg. constant sized ring signatures have been proposed by Bose et. al. [12] but neither implementation nor detailed algorithm is available yet.

Chapter 8

Conclusion

This thesis addresses the problem of revealing of information to the service through usage of static identifiers. It takes the use case of EV charging and intends to protect the EVU's location information from the service providers (CSO, ER and DSO). A previous privacy preserving solution proposed the use of different DIDs in every transaction but required the generation and storage of a large number of DIDs. The goal of this thesis was to study the use of ring signatures to achieve better privacy with less number of DIDs.

Previous work in this field showed that location privacy in EV charging is a long-standing issue and various solution have been proposed such as usage of group signatures, ZKP, escrow, etc. However, most of the solutions were either centralised or resource-intensive and hence unsuitable for implementation on constrained devices. DIDs and VCs used in the thesis are still being developed hence relatively less research have been done with these technologies.

Based on the previous solution of using different DIDs, an efficient and secure design called the Multi-DID Design was developed for comparison. Various design choices were made such as the type of DIDs used, credential format, micropayments design, etc. The Multi-DID design was modified to create the Ring Signature Design, which uses ring signatures to create billing evidence such that it protects the exact location of the EVU and yet allow other parties to prove the location of the transaction at a district level granularity. The EV and CS prototypes for both designs were implemented on Android devices which communicate using Bluetooth Low Energy. An assumption on the number of charging transactions was done based on data for the country of Norway.

The ring signature performance was measured on various fronts: storage space, processing time, user experience, DID and credential requirement and billing evidence verification. Measurements showed that ring signatures

consume both more storage space and processing power as the ring size increases. Although ring signatures require fewer credentials and enable DID reuse, the resource consumption to generate and verify ring signature more than outweighs the savings from credentials and DIDs. It was concluded that ring signatures do provide more privacy but consume significantly more resources.

Bibliography

[1] af Heurlin, L., et al. Authorization certificate based access control in embedded environments.

[2] Aftab, H., Gilani, K., Lee, J., Nkenyereye, L., Jeong, S., and Song, J. Analysis of identifiers on iot platforms. *Digital Communications and Networks* (2019).

[3] Aleksander Nowakowski. Android-nrf-connect. `https://github.com/NordicSemiconductor/Android-nRF-Connect`, 2019. [Online; accessed 26-July-2020].

[4] Ansey, R., Kempf, J., Berzin, O., Xi, C., and Sheikh, I. Gnomon: Decentralized identifiers for securing 5g iot device registration and software update. In *2019 IEEE Globecom Workshops (GC Wkshps)* (2019), IEEE, pp. 1–6.

[5] Antonino, A. A privacy-preserving approach to grid balancing using scheduled electric vehicle charging.

[6] Antonino, A., Biswas, S., Kortesniemi, Y., and Lagutin, D. Enhancing location privacy of electric vehicle in public charging systems. *Unpublished Manuscript* (2020).

[7] Au, M. H., Liu, J., Fang, J., Jiang, Z., Susilo, W., and Zhou, J. A new payment system for enhancing location privacy of electric vehicles. *Vehicular Technology, IEEE Transactions on 63* (01 2014), 3–18.

[8] Bender, A., Katz, J., and Morselli, R. Ring signatures: Stronger definitions, and constructions without random oracles. In *Theory of Cryptography Conference* (2006), Springer, pp. 60–79.

[9] Bernstein, D. J. Curve25519: new diffie-hellman speed records. In *International Workshop on Public Key Cryptography* (2006), Springer, pp. 207–228.

[10] Bernstein, D. J., Duif, N., Lange, T., Schwabe, P., and Yang, B.-Y. High-speed high-security signatures. *Journal of cryptographic engineering 2*, 2 (2012), 77–89.

[11] Biswas, S., and Antonino, A. Ev and cs android application with indy and bluetooth: indy-android-ev-cs. `https://github.com/SOFIE-project/indy-android-ev-cs`, 2020. [Online; accessed 24-July-2020].

[12] Bose, P., Das, D., and Rangan, C. P. Constant size ring signature without random oracle. In *Australasian Conference on Information Security and Privacy* (2015), Springer, pp. 230–247.

[13] Camenisch, J., and Stadler, M. Efficient group signature schemes for large groups. In *Annual International Cryptology Conference* (1997), Springer, pp. 410–424.

[14] Cantor, S., and Scavo, T. Shibboleth architecture. *Protocols and Profiles 10* (2005), 16.

[15] Cirillo, F., Solmaz, G., Berz, E. L., Bauer, M., Cheng, B., and Kovacs, E. A standard-based open source iot platform: Fiware. *IEEE Internet of Things Magazine 2*, 3 (2019), 12–18.

[16] Cohen, G., and Steele, O. Jcs ed25519 signature 2020. `https://identity.foundation/JcsEd25519Signature2020`, 2020. [Online; accessed 26-July-2020].

[17] Collotta, M., Pau, G., Talty, T., and Tonguz, O. K. Bluetooth 5: A concrete step forward toward the iot. *IEEE Communications Magazine 56*, 7 (2018), 125–131.

[18] Daniel Esteban. Android apps with gomobile. `https://github.com/conejoninja/MyGoApplication`, 2017. [Online; accessed 26-July-2020].

[19] Dave Longley and Manu Sporny. Linked data proofs 1.0. `https://w3c-ccg.github.io/ld-proofs`, 2020. [Online; accessed 4-July-2020].

[20] David Chadwick et al. Verifiable credentials implementation guidelines 1.0. `https://w3c.github.io/vc-imp-guide`, 2019. [Online; accessed 21-June-2020].

[21] DEDIS. Advanced crypto library for the go language. `https://github.com/dedis/kyber`, 2020. [Online; accessed 26-July-2020].

[22] Erik Lorentzen, Petter Haugneland, Christina Bu, Espen Hauge. Charging infrastructure experiences in norway - the worlds most advanced ev market. `https://elbil.no/wp-content/uploads/2016/08/EVS30-Charging-infrastrucure-experiences-in-Norway-paper.pdf`, 2017. [Online; accessed 16-June-2020].

[23] Frank Denis. A modern, portable, easy to use crypto library. `https://github.com/jedisct1/libsodium`, 2020. [Online; accessed 31-May-2020].

[24] Fries, S., and Falk, R. Electric vehicle charging infrastructure: Security considerations and approaches.

[25] Hardman, D., Kulic, D., Gudkov, V., and Lodder, M. Aries rfc 0050: Wallets. `https://github.com/hyperledger/aries-rfcs/blob/master/concepts/0050-wallets/README.md`, 2018. [Online; accessed 26-July-2020].

[26] Hartog, K. D., Curran, S., Curren, S., and Lodder, M. Aries rfc 0019: Encryption envelope. [Online; accessed 26-July-2020].

[27] J. Bernstein, D., and Lange, T. ebacs: Ecrypt benchmarking of cryptographic systems. `https://bench.cr.yp.to`, 2019. [Online; accessed 7-July-2020].

[28] Jean-Philippe Aumasson, Samuel Neves, Zooko Wilcox-O'Hearn, Christian Winnerlein. Blake 2x. `https://blake2.net/blake2x.pdf`, 2020. [Online; accessed 7-July-2020].

[29] Jones, M., Bradley, J., and Sakimura, N. Json web signature (jws). *Internet Requests for Comments, RFC 7515* (2015).

[30] Kortesniemi, Y., Lagutin, D., Elo, T., and Fotiou, N. Improving the privacy of iot with decentralised identifiers (dids). *Journal of Computer Networks and Communications 2019* (2019).

[31] Kursawe, K., Danezis, G., and Kohlweiss, M. Privacy-friendly aggregation for the smart-grid. In *International Symposium on Privacy Enhancing Technologies Symposium* (2011), Springer, pp. 175–191.

[32] Langheinrich, M. Privacy by design-principles of privacy-aware ubiquitous systems. In *International conference on Ubiquitous Computing* (2001), Springer, pp. 273–291.

[33] Li, H., Dán, G., and Nahrstedt, K. Portunes+: Privacy-preserving fast authentication for dynamic electric vehicle charging. *IEEE Transactions on Smart Grid 8*, 5 (2016), 2305–2313.

[34] Linux Foundation. Hyperledger indy project. `https://www.hyperledger.org/use/hyperledger-indy`, 2020. [Online; accessed 26-July-2020].

[35] Lundkvist, C., Heck, R., Torstensson, J., Mitton, Z., and Sena, M. Uport: A platform for self-sovereign identity. *URL: https://whitepaper. uport. me/uPort_ whitepaper_DRAFT20170221. pdf* (2017).

[36] Lux, Z. A., Thatmann, D., Zickau, S., and Beierle, F. Distributed-ledger-based authentication with decentralized identifiers and verifiable credentials. *arXiv preprint arXiv:2006.04754* (2020).

[37] Malina, L., Hajny, J., Dzurenda, P., and Ricci, S. Lightweight ring signatures for decentralized privacy-preserving transactions. pp. 692–697.

[38] Meiklejohn, S. An exploration of group and ring signatures.

[39] NOBIL. Publicly available charging points in norway. `https://info.nobil.no/statistikk`, 2020. [Online; accessed 4-July-2020].

[40] Noether, S. Ring signature confidential transactions for monero. *IACR Cryptology ePrint Archive 2015* (2015), 1098.

[41] Nordic Semiconductor. Ble on android v1.0.1. `https://devzone.nordicsemi.com/nordic/nordic-blog/b/blog/posts/what-to-keep-in-mind-when-developing-your-ble-andr`, 2016. [Online; accessed 26-July-2020].

[42] Oskar Deventar et al. Peer did method specification: Blockchain-independent decentralized identifiers. `https://openssi.github.io/peer-did-method-spec`, 2020. [Online; accessed 28-June-2020].

[43] Potzmader, K., Winter, J., Hein, D., Hanser, C., Teufl, P., and Chen, L. Group signatures on mobile devices: Practical experiences. In *Trust and Trustworthy Computing* (Berlin, Heidelberg, 2013),

M. Huth, N. Asokan, S. Čapkun, I. Flechais, and L. Coles-Kemp, Eds., Springer Berlin Heidelberg, pp. 47–64.

[44] Raikwar, M., Gligoroski, D., and Kralevska, K. Sok of used cryptography in blockchain. *IEEE Access 7* (2019), 148550–148575.

[45] Reed, D., Sporny, M., and Sabadello, M. Decentralized identifiers (dids) v1.0 technical report. Technical report, W3C, 2020. `https://www.w3.org/TR/did-core/`.

[46] Rivest, R., and Shamir, A. Payword and micromint: Two simple micropayment schemes. *URL http://people. csail. mit. edu/rivest/RivestShamir-mpay. pdf* (2001).

[47] Rivest, R. L., Shamir, A., and Tauman, Y. How to leak a secret. In *Advances in Cryptology — ASIACRYPT 2001* (Berlin, Heidelberg, 2001), C. Boyd, Ed., Springer Berlin Heidelberg, pp. 552–565.

[48] Robert Duffer. Why norway leads the world in electric vehicle adoption. `https://www.greencarreports.com/news/1123160_why-norway-leads-the-world-in-electric-vehicle-adoption`, 2020. [Online; accessed 1-July-2020].

[49] Rundgren, A., Jordan, B., and Erdtman, S. Json canonicalization scheme (jcs). `https://tools.ietf.org/html/draft-rundgren-json-canonicalization-scheme-13`, 2019. [Online; accessed 26-July-2020].

[50] Ryan West, Daniel Bulhm, Matthew Hailstone, Stephen Curran, Sam Curren. Aries rfc 0023: Did exchange protocol 1.0. `https://github.com/hyperledger/aries-rfcs/tree/master/features/0023-did-exchange`, 2019. [Online; accessed 4-July-2020].

[51] Sakimura, N., Bradley, J., Jones, M., De Medeiros, B., and Mortimore, C. Openid connect core 1.0. *The OpenID Foundation* (2014), S3.

[52] Simon Josefsson and Ilari Liusvaara. Edwards-curve digital signature algorithm (eddsa). `https://tools.ietf.org/html/rfc8032`, 2017. [Online; accessed 8-July-2020].

[53] Smith, S. M., and Khovratovich, D. Identity system essentials. *Evemyrn, Mar 29* (2016), 16.

[54] Sporny, M., Longley, D., and Others. Json-ld 1.1: A json-based serialization for linked data. `https://www.w3.org/TR/json-ld/`, 2020. [Online; accessed 26-July-2020].

[55] Sporny, Manu and et al. Verifiable credentials data model 1.0: Expressing verifiable information on the web. `https://www.w3.org/TR/vc-data-model`, 2019. [Online; accessed 26-July-2020].

[56] Stadler, I. Power grid balancing of energy systems with high renewable energy penetration by demand response. *Utilities Policy 16*, 2 (2008), 90 – 98. Sustainable Energy and Transportation Systems.

[57] Statistics Norway. Ev fleet. `https://www.ssb.no/transport-og-reiseliv/statistikker/bilreg`, 2020. [Online; accessed 4-July-2020].

[58] Steiner, J. G., Neuman, B. C., and Schiller, J. I. Kerberos: An authentication service for open network systems. In *Usenix Winter* (1988), Citeseer, pp. 191–202.

[59] Stuart Kent. Bluetooth low energy on android. `https://www.stkent.com/2017/09/18/ble-on-android.html`, 2017. [Online; accessed 26-July-2020].

[60] Swetina, J., Lu, G., Jacobs, P., Ennesser, F., and Song, J. Toward a standardized common m2m service layer platform: Introduction to onem2m. *IEEE Wireless Communications 21*, 3 (2014), 20–26.

[61] Terl. An android implementation of the libsodium cryptography library. for the lazy dev. `https://github.com/terl/lazysodium-android`, 2020. [Online; accessed 31-May-2020].

[62] Tobin, A., and Reed, D. The inevitable rise of self-sovereign identity. *The Sovrin Foundation 29*, 2016 (2016).

[63] Wang, Z. A privacy-preserving and accountable authentication protocol for iot end-devices with weaker identity. *Future Generation Computer Systems 82* (2018), 342–348.

[64] Wikipedia contributors. List of municipalities of norway — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=List_of_municipalities_of_Norway&oldid=946960101`, 2020. [Online; accessed 17-June-2020].

[65] Wood, G., et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper 151*, 2014 (2014), 1–32.

[66] Yang, Y., Cai, H., Wei, Z., Lu, H., and Choo, K.-K. R. Towards lightweight anonymous entity authentication for iot applications. In *Information Security and Privacy* (Cham, 2016), J. K. Liu and R. Steinfeld, Eds., Springer International Publishing, pp. 265–280.

Appendix A

Credentials and Presentations

This section contains the structure of the credentials and presentations used by the EV and CS in the Multi-DID and Ring Signature Design. As mentioned before, the credentials and presentations use the Json-Linked Data format and the signature suite used is *JcsEd25519Signature2020*. In the CS presentation for the Ring Signature Design, the signature suite is denoted as *Ed25519RingSignature*.

```
1
2  {
3    "@context": [
4      "https://www.w3.org/2018/credentials/v1",
5      "https://www.w3.org/2020/credentials/ev-info/v1"
6    ],
7    "id": "https://www.w3.org/2020/credentials/ev-info",
8    "type": [
9      "VerifiableCredential",
10     "EVChargingCredential"
11   ],
12   "credentialSubject": {
13     "id": "EV.did@EV:CS"
14   },
15   "issuer": "ER.did",
16   "issuanceDate": "2020-12-31T00:00:00Z",
17   "expirationDate": "2020-12-31T23:59:59Z",
18   "proof": {
19     "type": "JcsEd25519Signature2020",
20     "created": "2020-12-31T00:00:00Z",
21     "proofPurpose": "assertionMethod",
22     "verificationMethod": "ER.did#key1",
23     "signatureValue": "eyJhbGciOiJ...IsImI"
24   }
25 }
```

Listing A.1: Structure of the EV credential issued to a EV by the ER. The EV credential structure is common to both the designs. The EV DID authorised by the credential is represented by the field *credentialSubject*. The ER public DID and the verification key index provided enables other parties to verify the credential.

```
1
2  {
3    "@context": [
4      "https://www.w3.org/2018/credentials/v1",
5      "https://www.w3.org/2020/credentials/cs-info/v1"
6    ],
7    "id": "https://www.w3.org/2020/credentials/cs-info",
8    "type": [
9      "VerifiableCredential",
10     "CSInfoCredential"
11   ],
12   "credentialSubject": {
13     "id": "CS.did@EV:CS",
14     "district": "1"
15   },
16   "issuer": "CSO.did",
17   "issuanceDate": "2018-03-12T07:10:31Z",
18   "expirationDate": "2024-12-31T23:59:59Z",
19   "proof": {
20     "type": "JcsEd25519Signature2020",
21     "created": "2018-03-12T07:10:31Z",
22     "proofPurpose": "assertionMethod",
23     "verificationMethod": "CSO.did#key1",
24     "signatureValue": "JG7JcHzDi...DzrBar"
25   }
26 }
```

Listing A.2: Structure of the CS credential issued to a CS by the CSO in the Multi-DID Design. The CS DID authorised and the CS district location is represented by the field *credentialSubject*. The CSO public DID and the verification key index provided enables other parties to verify the credential.

```
1
2  {
3    "@context": [
4      "https://www.w3.org/2018/credentials/v1",
5      "https://www.w3.org/2020/credentials/cs-info/v1"
6    ],
7    "id": "https://www.w3.org/2020/credentials/cs-info",
8    "type": [
```

```
    "VerifiableCredential",
    "CSInfoCredential"
  ],
  "credentialSubject": {
    "ids": [
      "CS1.did@EV:CS",
      "CS2.did@EV:CS",
      .
      .
      .
      "CS49.did@EV:CS",
      "CS50.did@EV:CS"
    ],
    "district": "1"
  },
  "issuer": "CSO.did",
  "issuanceDate": "2018-03-12T07:10:31Z",
  "expirationDate": "2024-12-31T23:59:59Z",
  "proof": {
    "type": "JcsEd25519Signature2020",
    "created": "2018-03-12T07:10:31Z",
    "proofPurpose": "assertionMethod",
    "verificationMethod": "CSO.did#key1",
    "signatureValue": "JG7JcHzDi...DzrBar"
  }
}
```

Listing A.3: Structure of the CS credential issued to a CS by the CSO in the Ring Signature Design. The credential authorises a set of CS DIDs belonging to CSs in the district specified by the field *credentialSubject*. The CSO public DID and the verification key index provided enables other parties to verify the credential.

```

{
  "presentation": {
    "@context": [
      "https://www.w3.org/2018/credentials/v1",
      "https://www.w3.org/2018/credentials/examples/v1"
    ],
    "id": "urn:uuid:18E15106-E6DC-4EB5-8DEB-BFBFAC1C7A7A",
    "type": [
      "VerifiablePresentation"
    ],
    "ev-did": "EV.did@EV:CS",
    "proof": [
      {
        "type": "Ed25519RingSignature",
```

```
16          "created": "2020-08-11T17:22:41Z",
17          "proofPurpose": "assertionMethod",
18          "nonce": 1597155761,
19          "verificationMethod": "CS.credential.id",
20          "signatureValue": "edqw331Si...gFWd3kLas"
21        }
22      ]
23    }
24  }
```

Listing A.4: Structure of the CS presentation in the Ring Signature Design, created by the CS for the EV as proof of possession of the CS Credential and the ownership of one of the CS DIDs mentioned in the credential. The proof field contains a Ring Signature created by the key associated with the CS DID. The signature is created on the EV DID which proves the intended recipient of the presentation. The verification method is defined as the CS Credential Id which specifies the signer anonymity set. The CS presentation in the Multi-DID Design has the same structure but with the JcsEd25519Signature2020 suite and CS DID as the verification method.

```
1
2   {
3     "presentation": {
4       "@context": [
5         "https://www.w3.org/2018/credentials/v1",
6         "https://www.w3.org/2018/credentials/examples/v1",
7
8       ],
9       "id": "urn:uuid:13CB2439-CA8F-46FB-95B8-3F6F0642B9B8",
10      "type": [
11        "VerifiablePresentation",
12
13      ],
14      "commitment": {
15        "cs-signature": {
16          "ev-did": "EV.did@EV:CS",
17          "proof": [
18            {
19              "type": "Ed25519RingSignature",
20              "created": "2020-08-11T17:22:41Z",
21              "proofPurpose": "assertionMethod",
22              "nonce": 1597155761,
23              "verificationMethod": "CS.credential.id",
24              "signatureValue": "edqw331Si...gFWd3kLas"
25            }
26          ]
27        },
28        "w0": 527436582692,
```

```
29        "alg": "SHA-256",
30        "n": 50,
31        "p": 0.20
32      },
33      "proof": [
34        {
35          "type": "JcsEd25519Signature2020",
36          "created": "2020-12-31T09:20:12Z",
37          "proofPurpose": "assertionMethod",
38          "nonce": 1597155761,
39          "verificationMethod": "EV.did@EV:CS",
40          "signatureValue": "eyJ0eXAi...gFWFOEjXk"
41        }
42      ]
43    }
44  }
```

Listing A.5: Structure of the Payment Commitment in the Ring Signature Design, created by the EV for the CS as proof of possession of the EV Credential and to commit the payment details. The *cs-signature* field inside *commitment* contains the EV DID and Ring Signature from the CS presentation. The proof field for the Payment Commitment contains a regular signature by the key associated with EV DID to authenticate the commitment. The CS presentation in the Multi-DID Design has the same structure except for the *cs-signature* field which is replaced a with *cs-did* field and contains just the CS DID instead of a Ring Signature