EUROPEAN CENTRAL BANK

EUROSYSTEM

BANK OF JAPAN

STELLA – joint research project of the European Central Bank and the Bank of Japan

Balancing confidentiality and auditability in a distributed ledger environment

February 2020

The analysis and results presented in this report are not geared towards replacing or complementing existing arrangements, which include central bank-operated payment systems. Legal and regulatory aspects are outside the scope of the project.

EUROPEAN CENTRAL BANK
EUROSYSTEM

STELLA

BANK OF JAPAN

Project Stella

Balancing confidentiality and auditability in a distributed ledger environment

Executive summary

Over the past years a number of solutions have been developed to cater for the privacy and confidentiality aspects which arise as a result of the sharing of transaction information in distributed ledgers. These solutions focus, for example, on limiting access to information by unauthorised parties and are generally known as privacy-enhancing technologies/techniques (PETs).

The use of PETs may pose challenges, however, when third parties need to view and interpret the transaction for auditing purposes. To ensure accountability, the level of auditability aimed for in payment and settlement systems based on distributed ledger technologies (DLT) should be similar to that in centralised systems. This is applicable regardless of the different types of settlement assets, including stablecoins, central bank digital currency (CBDC) and others. Against this background, Stella phase 4 explores through conceptual studies and practical experimentation how confidentiality and auditability could be balanced in a distributed ledger environment. Specifically, it assesses the way in which PETs would ensure confidentiality as well as the arrangements that accommodate effective auditing for transactions in a financial market infrastructure (FMI) based on DLT.

Stella phase 4 divides PETs into three categories based on the underlying concepts for making transaction information confidential to unauthorised third parties. Segregating PETs ensure that each participant only has visibility into a subset of all transactions conducted in the network. Hiding PETs make use of cryptographic techniques to prevent third parties from interpreting transaction details. Unlinking PETs make it difficult to determine transacting relationships from the information recorded on the shared ledger.

Stella phase 4 proposes that the auditability of transactions in a DLT-based FMI using PETs can be assessed from the following key perspectives: accessibility to necessary information, reliability of the obtained information and efficiency of the auditing process. Accessibility refers to whether the auditor can access the information it needs to conduct auditing activities. Accessibility may be ensured if the auditor receives the information either from trusted sources (i.e. central components of the DLT system or credible third parties which provide particular functions for enabling PETs and possess necessary information) or from identifiable participants.

Reliability indicates whether the auditor can be certain that the original transaction information can be acquired using the obtained information. Reliability may be ensured if the auditor receives the necessary information from trusted sources or if it can use information recorded on the ledger to verify the correctness of the obtained information. The efficiency of the auditing process, which could be measured by the consumption of resources, is also considered since it would affect the feasibility of the process.

The assessment of the auditability of each PET setup based on the above perspectives finds that the following arrangements would contribute to effective auditing: (i) the auditor obtains the necessary information from trusted sources or (ii) the auditor obtains the necessary information from identifiable participants and has the means of verifying the correctness of the obtained information using information recorded on the ledger, and the entire process could be conducted without consuming excessive resources.

Stella phase 4 raises points to be considered further when expanding the discussion on balancing confidentiality and the auditability of transactions for practical application. First, it notes that the reliance on a trusted source could pose single point of failure risks for the network. Second, when multiple PETs are used in combination, there could be a trade-off between enhancing confidentiality and effective auditability. Third, when the model accommodates multiple payment and settlement systems as well as multi-tiered payment systems, it would be necessary to coordinate different standards and processes between systems. Last but not least, the inclusion of end-users may increase the complexity of managing the confidentiality of end-user information and necessitate the creation of appropriate standards to determine the transactions to be audited.

Contents

1 Introduction

Over the last few years, the European Central Bank (ECB) and the Bank of Japan (BOJ) have jointly explored the opportunities and challenges of distributed ledger technologies (DLT) for financial market infrastructures (FMI) in Project Stella. Launched in December 2016, Project Stella aims to contribute to the wider debate on the possible usage of DLT in the field of payments and financial market infrastructures via experimental work and conceptual studies. Previous phases of Project Stella[1] arrived at quantitative results on performance and resilience testing around DLT-based market infrastructures (September 2017) and explored the synchronisation mechanisms between different ledgers – including those between DLT-based and centralised ledgers – and asset classes (March 2018 and June 2019).

Progress has been made by the blockchain community in improving DLT for implementation in various use cases. There are also learnings from initiatives by various entities to create DLT-based platforms for payments and securities settlements. In this context, a number of solutions have been developed to cater for the privacy and confidentiality aspects which arise as a result of sharing transaction information on distributed ledgers. These solutions focus, for example, on limiting access to information by unauthorised parties and are generally known as privacy-enhancing technologies/techniques (PETs).[2]

To ensure accountability on DLT-based FMIs, it is necessary to have an arrangement in place in which authorised third parties can understand details of transactions to the same extent as in existing FMIs. This becomes a challenge, however, when PETs are applied to transactions since they could prevent third parties from viewing and interpreting transaction information. This report uses the term “auditability” to refer to the understanding of transaction information by the authorised third parties, or the degree to which a given environment allows an authorised entity to audit confidential transaction information by viewing and interpreting the information.

Explorations of privacy and confidentiality of transaction information in a distributed ledger environment have been made publicly available by the central bank

---

1 See *Payment systems: liquidity saving mechanisms in a distributed ledger environment*, ECB and BOJ, September 2017; *Securities settlement systems: delivery-versus-payment in a distributed ledger environment*, ECB and BOJ, March 2018; *Synchronised cross-border payments*, ECB and BOJ, June 2019.

2 There are wide-ranging definitions of PETs. See *Readiness analysis for the adoption and evolution of privacy enhancing technologies*, European Union Agency for Network and Information Security (ENISA), March 2016.

community.[3] It appears, however, that only limited research and experimentation is available with regards to the auditability of transaction information to which PETs have been applied (hereafter referred to as confidential transaction information).

Against this background, Stella phase 4 aims to offer insight into striking a balance between confidentiality and auditability of transaction information. More specifically, it introduces and systematically groups several PETs used in a DLT environment and assesses whether confidential transaction information can be effectively audited by an authorised entity in the DLT network.[4]

Chapter 2 outlines an abstract and hypothetical FMI model in a DLT environment on which the analysis is based. Chapter 3 introduces a selection of PETs used in the DLT context, explains the basic nature of PETs which attempt to enhance confidentiality and offers a categorisation. Chapter 4 proposes perspectives for assessing auditability and then assesses whether confidential transaction information could be audited effectively. Experiments which supported the analysis are outlined in Chapter 5.

2 Abstract FMI model based on DLT

This chapter introduces an abstract FMI model based on DLT on which PETs are applied. The DLT-based FMI model assumes that a group of entities form a network in which transaction information is recorded and shared in a decentralised manner (Figure 1). This model is in contrast to the existing approach where transaction information is recorded, stored and shared based on a centralised FMI model (Figure 2). Under the DLT-based model, each participating entity operates its own DLT node, through which transactions are processed and transaction information is stored and viewed.

For the purpose of the report, it is assumed that there also exists an entity or a group of entities which are authorised to audit transactions by viewing and interpreting the

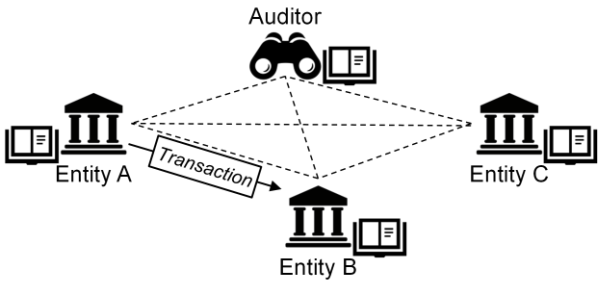
---

3 See *Distributed ledger technical research in Central Bank of Brazil*, Central Bank of Brazil, August 2017; *Project Jasper: a Canadian experiment with distributed ledger technology for domestic interbank payments settlement*, Bank of Canada, Payments Canada and R3, September 2017; *Project Ubin Phase 2: re-imagining interbank real-time gross settlement system using distributed ledger technologies*, Monetary Authority of Singapore and the Association of Banks in Singapore, November 2017; *Chain – fintech proof of concept*, Bank of England, April 2018; *Project Khokha: exploring the use of distributed ledger technology for interbank payments settlement in South Africa*, South African Reserve Bank, June 2018; and *Beyond theory: getting practical with blockchain*, Federal Reserve Bank of Boston, February 2019.

4 The joint research was conducted by Dirk Bullmann (ECB team leader), Andrej Bachmann, Diego Castejón Molina, Cedric Humbert, Austeja Sostakaite and Naisa Tussi from the ECB, with contributions from Giuseppe Galano (Banca d'Italia), Kurt Alonso (Directorate General Information Systems, ECB); and by Michinobu Kishi (BOJ team leader), Takeshi Yamada, Tetsuro Matsushima, Masashi Hojo and Amika Matsui from the BOJ, with contributions from Shuji Kobayakawa (Professor at Meiji University and Advisor to the BOJ Stella team).

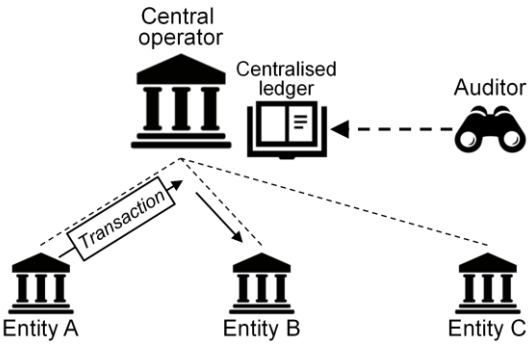
transaction information recorded on the ledger[5] ("auditor[s]")[6]. The focus of the report is on back-end arrangements and thus only covers transactions between participants. Accordingly, end-users (e.g. each participant's clients) do not appear in the model.

Figure 1
DLT-based FMI model



Note: Each participant stores relevant transaction information in its own ledger and shares the information with other participants.

Figure 2
Centralised FMI model



Note: The central operator owns a centralised ledger in which transaction information is stored. A participant's access to the centralised ledger is controlled by the central operator.

The DLT-based model is a permissioned network[7], also referred to as a private or restricted network, where all participants with granted access are expected to follow the terms and conditions (rules) of the network and fulfil their responsibilities. Participants are required to implement and use basic functions on their nodes that are compliant with the rules, and use a designated transaction format to process transactions. If the participants do not comply with the rules, they may be subject to sanctions, including losing access to the network, as well as face reputational risks.

It should be noted that the DLT-based model, for reasons of simplicity, does not cover system administrator roles such as gatekeeping and governance since these

---

5 It is theoretically possible to incorporate auditing into the transaction validation process. For example, see *Exploring anonymity in central bank digital currencies*, ECB, December 2019. However, this approach is not within the scope of this report.

6 If there are multiple auditors in the DLT network, there would need to be mechanisms in place to ensure that effective auditing is conducted. These can include: a) a mechanism to ensure that every piece of transaction information is auditable by at least one auditor; b) a mechanism that enables auditors to know which auditor is responsible for a given transaction; and c) a mechanism that enables auditors or other entities to share relevant information amongst themselves as necessary, for example, through a shared database between auditors.

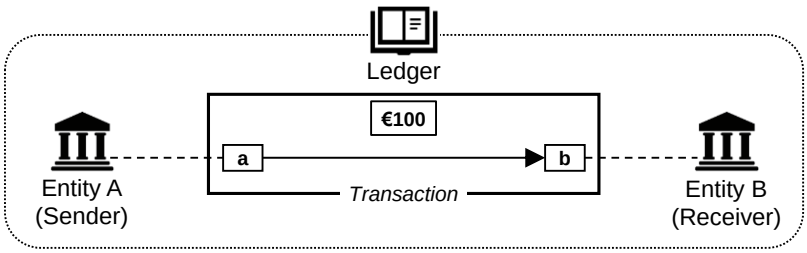
7 A permissioned network is a type of DLT network where an entity cannot participate without authorisation by other participants or a system administrator, if there is one.

are not the primary focus of this report. The role of a transaction validator could be assumed by participants or other authorised entities and is addressed in the report where relevant.

Moreover, for the sake of simplicity, transaction information only contains the transacting parties (participant identifiers, or addresses of the sender and receiver) and the transaction amount. While information on end-users or that related to smart contracts[8] could have enriched the DLT-based model, it is disregarded because its inclusion would not have materially impacted the main findings in Chapter 4. Figure 3 illustrates the transaction information on the ledger for a transaction where Entity A (sender) transfers €100 to Entity B (receiver).

Figure 3

Illustrative example of transaction information



Note: The dashed line denotes the relationship between addresses (a, b) and transacting parties (A, B) who use them.

Confidentiality in the abstract FMI model

Various terms and definitions exist in the current discussion on confidentiality or protection of transaction data. For the purpose of this report, the term “confidentiality” denotes a concept that, when in place, ensures that unauthorised third parties (denoted by Entity C in Figure 1 and Figure 2) are unable to view or interpret the transaction information (between Entities A and B). In this report, “view” refers to the existence of the transaction information being visible to third parties while “interpret” refers to a situation where third parties are able to not only view but also derive the actual value and/or identify the transacting parties.

In a centralised model (Figure 2), the confidentiality of transaction information could be ensured by a central operator which manages transaction information so that each participant can retrieve only the information to which it is granted access. That is, the central operator effectively prevents unauthorised third parties from viewing the particular transaction information. A similar level of confidentiality must be

[8] Smart contract is a way of transposing participants’ contractual obligations onto DLT platforms and ensures that provisions are automatically executed.

ensured in the DLT-based model (Figure 1), despite the fact that information is stored and shared in a decentralised manner.

Auditability in the abstract FMI model

In this report, auditability refers to the degree to which a given environment allows the auditor to conduct an effective audit – that is, to view and interpret transaction information to fulfil its responsibility.[9]

Auditability in a centralised model could be achieved if the central operator discloses transaction information upon a request from the auditor. This is possible since, in the centralised model, the information is not confidential towards the central operator.

A similar level of auditability should be aimed at in a DLT-based model, even without the presence of the central operator.[10] The auditor would run a DLT node in the network for this purpose, although there may be cases where the auditor accesses the ledger via participants in the network. However, use of PETs may pose a challenge to auditability as they aim to make the transaction information not viewable or, even if viewable, uninterpretable by unauthorised third parties and the auditor. Thus, achieving sufficient levels of both confidentiality and auditability is a challenge for some setups of PETs.

3 Privacy-enhancing technologies/techniques on DLT

A number of solutions have emerged to cater for privacy and confidentiality aspects which arise as a result of sharing transaction information on distributed ledgers. Research and projects increasingly focus on possible improvements to enhance the confidentiality of transaction information.

This chapter introduces some technologies and techniques which aim at enhancing the confidentiality of transaction information in DLT networks, or so-called PETs. In doing so, it sorts PETs into three categories based on fundamental approaches to enhancing confidentiality. It assumes that basic pseudonymisation[11] is already applied in DLT networks prior to the application of PETs. It should be noted that the PETs described in this report are not mutually exclusive and could be applied in combination to further enhance confidentiality.

---

9 In addition, the auditor may verify as necessary whether each participant is appropriately conducting know-your-customer (KYC) and anti-money laundering (AML) checks. To this end, each participant should have capabilities to share its customer information with the auditor by maintaining a means to link its own customers' identities with on-ledger account details. However, as these capabilities would likely be established outside the DLT network, they are outside the scope of this report.

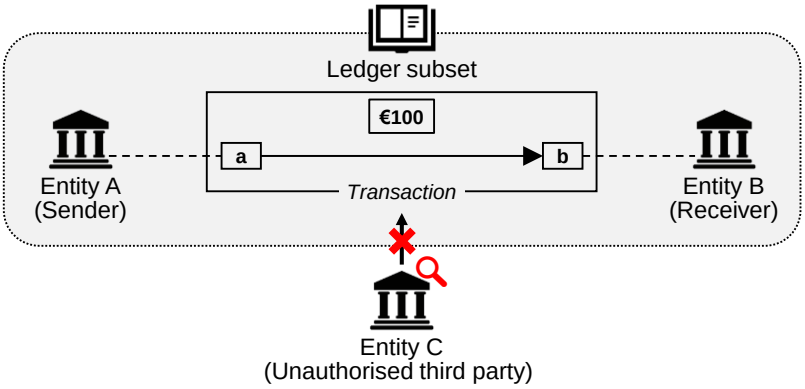
10 *Principles for financial market infrastructures*, Committee on Payments and Market Infrastructures, 2012, mentions confidentiality and auditability among the standards to which an FMI's information security objectives and policies should conform (see explanatory note 3.17.12).

11 Pseudonymisation is a means of disguising participants' personal data in such a manner that these data can no longer be attributed to the participants, and is commonly used in DLT networks. Network participants are usually associated with a unique identifier – a pseudonym. Although a pseudonym may not contain any information about the respective participant (generally it appears as a random string of letters and numbers), the use of pseudonyms alone does not offer a sufficient level of confidentiality.

3.1 Segregating PETs

Confidentiality could be achieved by designing the DLT network in a way that transaction information is segregated between participants and shared on a "need to know" basis (see Figure 4). When Segregating PETs are used, there is no shared ledger accessible by all participants containing the record of all individual transactions. Instead, each participant only has a record of a subset of all transactions (ledger subset). Thus, unauthorised third parties to a transaction (Entity C) cannot recognise the existence of the transaction (between Entities A and B).

Figure 4
Illustrative example of Segregating PETs



3.1.1 Segregating technique in Corda

The permissioned DLT platform Corda features a network design that effectively segregates transaction records. This is achieved through the way participants communicate within the network, so that transaction information is shared only among authorised participants.[12] In Corda, only predefined and identified participants can be part of a particular communication, while all the other participants in the network remain unaware of the transaction.

Although transaction information is segregated between participants, Corda networks have in place a network service, called a notary, which receives transaction information to ensure that funds are not spent twice. There are two types of notaries: validating notaries and non-validating notaries. Validating notaries receive all transaction information to validate transactions in addition to checking double-spending. Non-validating notaries receive only a part of the information, while the

[12] See *Corda: a distributed ledger*, M. Hearn, November 2016.

complete transaction information is shared between specific participants who check the validity of the transaction.[13]

3.1.2 Segregating technique in Hyperledger Fabric

The permissioned DLT platform Hyperledger Fabric offers its participants the option to segregate transaction information at the network level through a channel functionality. Such functionality effectively divides the network into subnetworks that have their own ledger subset, rather than maintaining a common ledger for all subnetworks.[14]

Participants must be authenticated and authorised by a network service to transact in and maintain a copy of the ledger of a specific channel. Therefore, a participant can only have access to transactions of channels in which it participates. Furthermore, in the most recent implementation of Hyperledger Fabric at the time of writing (version 1.4), the transaction information of every channel is sent to a network service for transaction ordering (ordering service).

3.1.3 Off-ledger payment channel

An off-ledger payment channel[15] is a solution that enhances confidentiality by enabling funds native to a particular network to be transacted outside the main network.[16] It allows participants to transact off-ledger without broadcasting each transaction to the entire network.

To establish (or open) a payment channel between two participants, one or both transacting parties deposit a certain amount of funds (€50 each from Entities A and B in Table 1), usually by escrowing it in a shared, temporary account on the ledger. Participants can conduct transactions off-ledger bilaterally by exchanging claims to the deposited funds. When the channel is closed, the deposited funds are split between the parties according to the final claim. Information on the opening and closing transactions is recorded on the shared ledger, enabling third parties to see the pseudonyms of the sender and receiver, as well as the net amount transacted in a particular channel (Entity A received €30 from Entity B). However, individual off-ledger transactions are not visible to other participants.

Multiple bilateral payment channels between several participants may form a payment channel network, whereby two participants without a bilateral payment

---

13 Furthermore, in Corda, participants who validate transactions may become aware of the confidential information of past transactions. To prevent this, additional techniques such as chain snipping can be implemented.

14 https://hyperledger-fabric.readthedocs.io/en/release-1.4/channels.html

15 For a more detailed description of payment channels, see *Synchronised cross-border payments*, ECB and BOJ, June 2019.

16 An alternative way of enhancing confidentiality by conducting transactions outside the main network is the concept of sidechains.

channel can transact by relaying their transaction through intermediate participants. If the intermediate participant is a central party that maintains a bilateral payment channel with all other participants and is in charge of relaying their off-ledger transactions, it essentially becomes a payment channel hub. As regards to confidentiality, however, a payment channel hub may see transaction information (i.e. the amount and transacting parties) when relaying the transaction.

Off-ledger payment channel setups are implemented on public blockchains such as Bitcoin and Ethereum (Lightning and Raiden networks, respectively).

Table 1

Illustrative example of transaction information from the viewpoint of Entity C

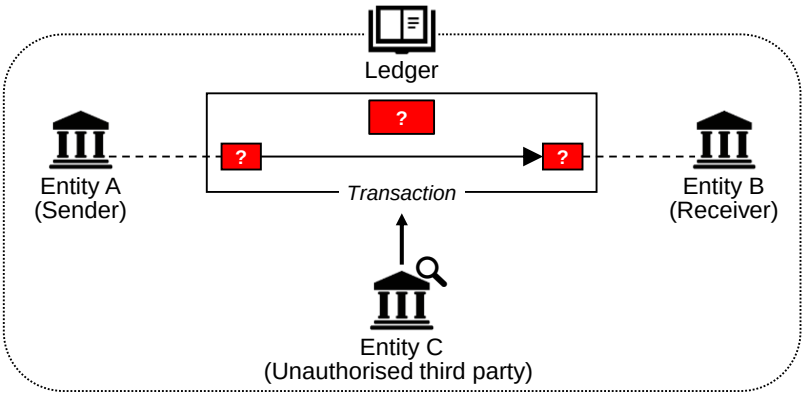
Visible information (On-ledger transactions)	Invisible information (Off-ledger transactions)
Opening transaction A→escrow account (€50) B→escrow account (€50)	
	A→B (€20)
	B→A (€40)
	B→A (€10)
Closing transaction escrow account→A (€80) escrow account→B (€20)	

3.2 Hiding PETs

PETs can also be used to enhance confidentiality at transaction level when transaction information is not segregated and participants share a single ledger which contains the record of every transaction. Despite all participants being able to view every transaction within the network, confidentiality could be enhanced by using various cryptographic techniques to prevent unauthorised third parties (Entity C) from interpreting transaction details, thereby effectively hiding them (Figure 5).

Figure 5

Illustrative example of Hiding PETs



3.2.1 Quorum's private transaction

The network of DLT platform Quorum[17] allows two types of transactions – public and private. Private transactions is an optional feature of Quorum enabling participants to hide their transaction information from unauthorised third parties.[18] This is enabled via the pre-configuration of transactions – prior to executing private transactions, participants designate parties to privately transact with. Information of private transactions is stored in private ledgers of the designated parties while the public ledger records the hash value[19] of the transaction information as well as the sender information. This allows the execution of private transactions without unauthorised third parties having the possibility to fully interpret the transaction information.

3.2.2 Pedersen commitment

Pedersen commitment is a type of cryptographic primitive that allows a sender to create a commitment to an amount and share the commitment instead of the amount itself.[20] The commitment is created using parameters defined by the network as well as those chosen by the sender.

Transacting parties could use Pedersen commitment to replace the transaction amount on the shared ledger with a commitment uninterpretable to third parties. Meanwhile, the sender and receiver information remains interpretable. Pedersen

---

17 Quorum is based on Ethereum, which is a public blockchain platform but is designed for processing transactions within a permissioned network.

18 The transaction information of public transactions, on the other hand, is interpretable by all network participants.

19 Hash value in this case is returned by a one-way hash function from the input. It is impossible to derive the input from the hash value.

20 See *Non-interactive and information-theoretic secure verifiable secret sharing*, T. P. Pedersen, 1991.

commitment allows third parties to verify that the input and output amounts of a transaction are equivalent without revealing them. Pedersen commitment has the cryptographic attributes of perfectly hiding and computationally binding, which means that interpreting a commitment requires information about the parameters that were used to create the commitment and that the underlying amount cannot change. Details of Pedersen commitment can be found in Chapter 5 and the Annex.

3.2.3 Zero-knowledge proof

Zero-knowledge proof (ZKP) is a cryptographic method allowing a party to prove the possession of information without disclosing any information apart from the fact that it possesses the information. In DLT networks, ZKP is used by transacting parties to create confidential transactions, whose information could be verified without revealing it.[21] By recording confidential transaction information on the ledger, transacting parties could conduct transactions while making the information uninterpretable from unauthorised third parties.

Several kinds of implementation have been developed based on this method, in particular ones that allow the verification of transactions without requiring interaction between senders and other participants. Zero-knowledge Succinct Non-interactive ARgument of Knowledge (zk-SNARK) has been proposed for the efficient execution of such non-interactive ZKP implementations. This scheme requires a one-time setup by a trusted party using a secret parameter to generate two public parameters: proving and verification keys. The proving key is used by senders to share fully encrypted transactions, while the verification key can then be used to validate them.

Several DLT platforms support applications based on zk-SNARK, such as Ethereum and Quorum. Improvements of zk-SNARK are being proposed. For example, DIstributed Zero Knowledge[22] promises better scalability, while zero-knowledge Scalable Transparent ARgument of Knowledge[23] aims to provide transaction verification without relying on a trusted party's setup.

---

21 ZKP can also be used to hide specific parts of a transaction.

22 *DIZK: a distributed zero knowledge proof system*, H. Wu, W. Zheng, A. Chiesa, R. A. Popa and I. Stoica, 2018.

23 *Scalable, transparent, and post-quantum secure computational integrity*, E. Ben-Sasson, I. Bentov, Y. Horesh and M. Riabzev, March 2018.

Box: Smart contract confidentiality

This paper assumes only payment and settlement processes where participants execute transactions on the DLT-based FMI model. Smart contracts may be used to extend this model to various financial applications. Some of the PETs introduced in this chapter can be used to enhance the confidentiality of smart contracts' logic.

Merkelized Abstract Syntax Tree (MAST)[24] is another technique which could be used to hide the logic of a smart contract. The design of MAST enables a conditional tree structure where mutually exclusive conditions in the logic are set as branches. The details of branches remain hidden as a hash unless a particular condition is executed. When a condition is executed then only the corresponding branch is revealed.[25]

3.3 Unlinking PETs

PETs could be applied to sever the relationship between the sender/receiver information visible on the shared ledger and that of the actual transaction.[26] As illustrated in Figure 6, this could be done in two ways: (i) by unlinking the actual identity of the sender (Entity A) and/or receiver (Entity B) from the recorded pseudonym (a and b) or (ii) by unlinking the transacting relationship between the sender and receiver.[27] Thus, unauthorised third parties to a transaction (Entity C) can view the transaction information and interpret the amount but cannot determine the transacting relationships (that Entity A transacted with Entity B).

---

24 *Merkelized Abstract Syntax Trees*, J. Rubin, M. Naik and N. Subramanian, 2014.

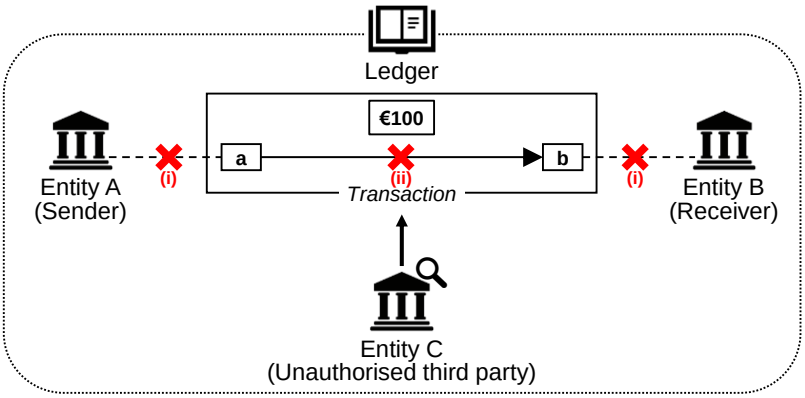
25 To further enhance confidentiality, particularly in cases of the mutually agreed execution of smart contracts, other techniques can be used. For example, Schnorr signatures can help aggregate participants' signatures, and thus hide the details of the executed branch. See *Simple Schnorr multi-signatures with applications to Bitcoin*, G. Maxwell, A. Poelstra, Y. Seurin and P. Wuille, May 2018.

26 This also assumes that transaction information is not segregated and participants share a single ledger which contains the record of every transaction.

27 In practice, unlinking transacting relationships is a weaker property since it could be possible to trace outgoing transactions from senders and incoming ones to receivers separately to recreate the link between them. See *Modelling unlinkability*, S. Steinbrecher and S. Köpsell, 2003.

Figure 6

Illustrative example of Unlinking PETs



3.3.1 One-time address

A participant could use different pseudonyms, or addresses, for every transaction (one-time address) to prevent its identity from being linked to other transactions that it is part of (pattern (i) in Figure 6). This technique is widely used in various schemes and projects.[28] While the technique enhances confidentiality, using one-time addresses may quickly result in a large number of addresses per participant, which entails the non-negligible complexity of managing them and their corresponding private keys.[29]

Deterministic wallet is one of the most common and efficient ways for participants to tackle this drawback. It allows participants to deterministically generate a number of addresses from a single starting point, alleviating the complexity of managing addresses. As there is no apparent relationship between individual addresses, it is difficult for third parties to link together transactions conducted using these addresses.

Among deterministic wallets, hierarchical deterministic (HD) wallet is the most practical, with the seed being used to create a master public/private key pair, from which all keys are derived in a hierarchical, tree-like manner. Other solutions include generating transaction specific keys from public keys and shared secrets, as done in CryptoNote derivatives.[30] HD wallet is described in more detail in Chapter 5 and the Annex.

[28] For example in Bitcoin, Ethereum and Libra.

[29] See Annex for an explanation of private keys.

[30] https://cryptonote.org/cns/cns006.txt

3.3.2 Mixing

Mixing is a technique that allows multiple participants to shuffle multiple transactions so that their transacting relationships cannot be linked (pattern (ii) in Figure 6). The resulting (mixed) transaction recorded on the shared ledger features multiple senders and receivers, which makes it challenging for unauthorised third parties to determine the original sets of transacting parties (see Table 2). In general, the larger the number of transactions mixed together, the higher the level of confidentiality due to the larger set of parties conducting transactions.

Mixing could be conducted through a centralised mixing service provider or on a peer-to-peer (P2P) basis. For participants to use a centralised mixing service provider, the provider may need to be trusted by participants as it would receive their original transaction information. Alternatively, in the case of P2P mixing, participants would not need to rely on the centralised mixing service provider; however, a challenge is to find other participants willing to transact at the same time.

The transaction amount is recorded on the ledger in interpretable form when mixing (both centralised and P2P) is used. Thus, transacting parties could be linked with one another if there is a limited number of transactions with the same amount. To overcome this drawback, mixing could be implemented in combination with techniques, such as Pedersen commitment, that conceals the amount. There are currently several mixing protocols and service providers offering mixing services.

Table 2

Illustrative example of transaction information before and after mixing from the viewpoint of Entity C

Before mixing		
Sender	**Amount**	**Receiver**
a	€100	b
d	€100	f
e	€100	g

After mixing		
Sender	**Amount**	**Receiver**
e	€100 each	b
a		g
d		f

3.3.3 Ring signature

Ring signature is a type of digital signature that can be used to prove that a signer belongs to a group of signers, without revealing the actual signer (pattern (ii) in Figure 6).[31]

The key property of ring signature on DLT networks is to enable a sender to gather multiple public keys of various participants (called ring members) and sign a transaction with its own private key, and the gathered public keys. A third party would

[31] The concept was introduced by *Group signatures*, D. Chaum and E. van Heyst, 1991 as well as *How to leak a secret*, R. L. Rivest, A. Shamir and Y. Tauman, 2001.

only know that one of the ring members signed the transaction, but would not be able to determine who.

However, similar to mixing, the transaction amount (as well as the pseudonym of the receiver) is recorded on the ledger in interpretable form when a ring signature is used (Table 3). The information on the amounts used as inputs could be used to single out the actual sender. Therefore, to enhance confidentiality, ring signatures are usually implemented in combination with techniques that conceal the amount, such as Pedersen commitment.

Table 3

Illustrative example of transaction information using a ring signature from the viewpoint of Entity C

Sender	Amount	Receiver
a d e	€100	b

3.4 Summary

Each PET's specificities have a different effect on the visibility and interpretability of transaction-related data. Table 4 provides an overview of whether transaction information can be viewed and interpreted by unauthorised third parties. It should be noted that using multiple PETs in combination may ensure a higher level of confidentiality.

Table 4

Whether transaction information can be viewed and interpreted by unauthorised third parties

Category	PETs	Transaction information		
		Sender	Receiver	Amount
Segregating	Segregating technique in Corda	No		No
	Segregating technique in Hyperledger Fabric	No		No
	Off-ledger payment channel	Yes		No*
Hiding	Quorum's private transaction	Yes	No	No
	Pedersen commitment	Yes		No
	Zero-knowledge proof (hiding sender, receiver and amount)	No		No
Unlinking	One-time address	No		Yes
	Mixing	No		Yes
	Ring signature	No	Yes	Yes

* Only the net transacted amount can be viewed and interpreted.

4 Auditability of confidential transaction information

When PETs are applied to enhance confidentiality of transactions in DLT systems, ensuring the auditability of transaction information can become a challenge. Building on the model presented in Chapter 2, this chapter assesses whether confidential transaction information could be audited effectively, with the aim of contributing to the discussion on achieving a balance between confidentiality and auditability in DLT-based networks.

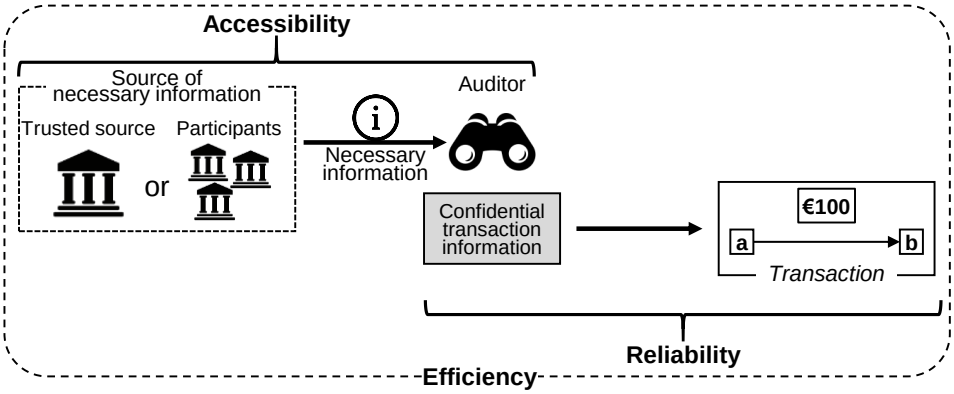
There could be a variety of solutions that implement PETs. Moreover, some of the PETs addressed in this report are still evolving. Therefore, the method of auditing and its effectiveness relies greatly on the features of the particular implementation of the PET in the network, and the results of the assessment for each PET are not conclusive.

In this chapter, key perspectives are proposed for assessing whether confidential transaction information could be audited effectively. Then, assessment of particular setups of PETs against these perspectives is provided. Additionally, further considerations for practical application are elaborated.

4.1 Three perspectives for assessing auditability

This report proposes three perspectives from which the auditability of confidential transaction information could be assessed. These perspectives correspond to the general flow of the auditing process, which is illustrated in Figure 7 and elaborated on below. The perspectives, in the order in which they are assessed, are: (i) accessibility to necessary information, (ii) reliability of the obtained information and (iii) efficiency of the auditing process. When the assessment results from these perspectives are positive for a particular DLT system setup, effective auditing could take place.

Figure 7
General flow of the auditing process and three perspectives



4.1.1 Accessibility to necessary information

The first perspective is accessibility to necessary information. It considers whether the auditor can obtain the information it needs to conduct auditing activities. When PETs are used, the auditor cannot view (i.e. when Segregating PETs are used) or interpret (i.e. when Hiding or Unlinking PETs are used) the transaction information. Consequently, additional information for auditing (i.e. necessary information) would need to be obtained via alternative data sources. These data sources from which the auditor receives the necessary information could either be "trusted sources" within the network or participants.

Trusted sources could be components of a DLT system which exist by design (e.g. a Corda notary) or credible third parties conducting the implementation of PETs (e.g. a centralised mixing service). They possess the necessary information which the auditor could use to interpret the transaction information with certainty. Trusted sources and participants could be required to cooperate with the auditor by the rules of the network with an enforcement mechanism, such as potential sanctions or loss of access to the network.

Accessibility to necessary information is ensured in the auditing processes that require trusted sources to submit necessary information to the auditor. In the absence of a trusted source, the auditor would need to rely on participants (especially transacting parties) as the primary source of necessary information. In most cases, participants are expected to adhere to the rules of the network by cooperating with the auditor. Nevertheless, there may be scenarios in which they fail to do so, requiring the auditor to identify the participants to enforce information retrieval. If the auditor can identify participants who possess the transaction information (hereafter referred to as identifiable participants), accessibility could be ensured through enforcement. When the auditor needs to rely on other participants, accessibility cannot be ensured.[32]

4.1.2 Reliability of obtained information

When the auditor can obtain the necessary information, the second perspective could be applied, which focuses on the reliability of the obtained information. Obtained information is considered reliable if the auditor can be certain that the original transaction information can be acquired using the obtained information.

Similar to the assessment of accessibility, when the auditor can rely on trusted sources, the necessary information obtained from them would be considered

---

[32] DLT systems could be designed in such a way that auditing is incorporated into the validation process. This would ensure accessibility as well as reliability regardless of the PET used. However, since this approach to auditability would result in real-time auditing and require an active operational role on the part of the auditor, it is outside of the scope of this report as described in Chapter 2. It is worth noting that there are proposals to design DLT systems in a way that the auditability of a transaction is verified in the validation process. See *Exploring anonymity in central bank digital currencies*, ECB, December 2019; and *Auditable zerocoin*, K. Naganuma, M. Yoshino, H. Sato, T. Suzuki, 2017.

reliable. In the case where there is no trusted source, the auditor would need to rely on cooperation from identifiable participants to obtain necessary information. The reliability of this information depends on whether there is a corresponding record on the shared ledger which the auditor could access and use to verify the correctness of the obtained information. When such a record is not available on the ledger, the reliability of the obtained information cannot be ensured.

4.1.3 Efficiency of auditing process

In addition to accessibility and reliability, considering the efficiency of the auditing process is essential to check whether auditing would be feasible. Efficiency could be measured by the consumption of resources (e.g. computational power, data storage and communication bandwidth) by the auditor, participants and other relevant parties (such as trusted sources). Auditing confidential transaction information would consume varying levels of resources based on the particular setup of the DLT system, and resource consumption may change over time according to available technology.

Conceptually, if the auditing process consumes an excessive amount of computational power or the network and auditing framework are set up in such a way that the auditor and participants must communicate for each transaction, the auditing process may not be considered sufficiently efficient. In extreme cases, such as when the auditor needs to find an exact value from a large number of potential values, auditing may even be considered infeasible. When the auditor obtains the necessary information from trusted sources, the auditing process could generally be conceived sufficiently efficient since the auditor would not need to communicate with participants. When the auditor obtains the necessary information from identifiable participants, efficiency could differ according to the particular setup of the network. The assessment of the three perspectives in auditing processes with each data source is summarised in Table 5.

Table 5

Assessment of three perspectives with different data sources

Data source	Accessibility	Reliability	Efficiency
Trusted sources	Yes	Yes	Yes
Identifiable participants	Yes	Depends on whether a record corresponding to the obtained information is on the ledger	Depends on particular setup and PET
Other participants	No	N/A	N/A

4.2 Assessment based on the perspectives

In this section, particular setups of PETs are assessed against the three perspectives described above. The discussion below is not meant to be comprehensive, in that it covers only selected PET setups in the permissioned DLT-based FMI model introduced in Chapter 2. Therefore, the assessment of auditing processes for PET setups under a different model may yield different results.

4.2.1 Segregating technique in Corda

Corda allows for several configurations, including the deployment of validating or non-validating notaries. By design, information for every transaction is shared with notaries. In addition, auditors could potentially run observer nodes[33] to which participants send their transaction information.

Validating notaries receive transaction information in interpretable form[34] to validate them; otherwise transactions are not accepted as effective. If the auditor receives the necessary information from validating notaries acting as trusted sources, it is expected that accessibility, reliability and a sufficient level of efficiency could be ensured.

Non-validating notaries receive transaction information in hidden form but store information on the sender of the information in interpretable form.[35] While the auditor must rely on cooperation from participants to submit transaction information, it could use the information stored in the non-validating notaries to verify the submitted transaction information and identify non-cooperative participants. Thus, when non-validating notaries are used in the auditing process, accessibility and reliability could be ensured, albeit with a lower level of efficiency than using validating notaries as trusted sources. To enhance the level of efficiency, the auditor could run an observer node to be used with non-validating notaries. This could be realised by configuring participants' node setups so that the observer node is included in all exchanges of transaction information.

---

33 There is a common idea of ensuring auditability by setting up a DLT node for the auditor in a way that necessary information is passed on to that node. Such a node is generally called supervisory node or observer node. For example, see https://docs.corda.net/tutorial-observer-nodes.html and *Beyond theory: getting practical with blockchain*, Federal Reserve Bank of Boston, February 2019.

34 https://docs.corda.net/key-concepts-notaries.html

35 Ibid.

4.2.2 Segregating technique in Hyperledger Fabric

In Hyperledger Fabric the information for all transactions conducted in channels is sent to the ordering services, from which the auditor may obtain the necessary information.[36] Ordering services could be regarded as trusted sources, and therefore accessibility, reliability and a sufficient level of efficiency could be ensured.

An alternative method for the auditor is to deploy observer nodes on the network. By configuring the network in a way that these nodes participate in all channels and the information for individual transactions is shared with them, the auditor could obtain the necessary information for auditing. In this case, accessibility and reliability of auditing could be ensured. With regards to efficiency, the auditor may bear some additional burden from managing observer nodes.

4.2.3 Off-ledger payment channel

The auditor needs the transaction information to be shared from transacting parties as payment channels only record the opening and closing transactions on the ledger and not individual transactions. Since the transacting parties are recorded in interpretable form for the opening and closing transactions, accessibility is ensured. However, while the auditor could check that the net amount of the submitted transaction information corresponds with the amount transacted through the opening and closing transactions, there is no way for the auditor to know if the submitted information on individual transactions is correct. Therefore, reliability cannot be ensured.[37]

If a payment channel hub exists in the network, it could act as a trusted source and share with the auditor the information on all individual transactions. In this case, accessibility and reliability would be ensured. Furthermore, this auditing arrangement would be considered sufficiently efficient as participants would not need to submit transaction information to the auditor.

4.2.4 Quorum's private transaction

With Quorum's private transactions, the sender information and the hash value of the transaction information are recorded on the public ledger. Although the auditor can interpret the sender information, it needs the sender to submit the transaction information, which it could verify with the hashed value recorded on the ledger.

---

[36] As previously mentioned, in the current version of Hyperledger Fabric, the transaction information is transmitted to the ordering service in interpretable form.

[37] In practice, transactions in a payment channel network are often relayed via multiple intermediary participants. This, however, may add complexity to auditing.

Therefore, this process could ensure accessibility and reliability. Nonetheless, this would pose a burden on the auditor and participants as additional communication is required to share necessary information, thereby negatively impacting efficiency.

A viable option to enhance efficiency may be the use of observer nodes. By configuring participants' node setups to send all transaction information to the observer node[38], the auditing arrangement would require no additional processes on the part of participants, which would ensure a sufficient level of efficiency.

4.2.5 Pedersen commitment

The use of Pedersen commitment leaves the identities of transacting parties interpretable to the auditor while the transaction amount is hidden as a commitment. To interpret the commitment, the auditor needs the transacting parties to share the parameter selected by the transacting party (blinding factor) and/or the amount.

Since Pedersen commitment does not hide the identity of the sender and the receiver, accessibility could be ensured. In addition, reliability could be ensured as the auditor can verify whether the obtained information is correct by calculating the commitment from the obtained information and comparing it with that recorded on the ledger. If the auditor receives the blinding factor and the amount, the computational burden for the auditor to calculate the commitment is minimal, so the process is sufficiently efficient. On the other hand, if the auditor only receives the blinding factor and the number of potential amounts is not limited, it may need to compute the amount with brute force, increasing its computational overhead significantly. So, from the perspective of efficiency, it is expected that the auditor would require that both the blinding factor and amount are shared. Experiments on auditing transaction information using Pedersen commitment are detailed in Chapter 5 and the Annex.

4.2.6 Zero-knowledge proof

There could be various implementations utilising ZKP. Assessment from the three perspectives would largely depend on the specific solution. When sender and receiver information is made confidential using ZKP, the auditor cannot identify the transacting parties from the information recorded on the shared ledger. Therefore, accessibility cannot be ensured. Although features to allow designated third parties to view the original transaction information (viewing keys) are being developed for

[38] By enabling the “always-send-to” option, it is possible to share information for every private transaction with observer nodes even if the sender does not specify them as parties to the transaction. See https://github.com/jpmorganchase/tessera/wiki/Configuration-overview.

certain applications, these features may not ensure accessibility as long as these keys need to be shared with the auditor by participants.[39]

4.2.7 One-time address

The technique allows a different pseudonymous address to be used for every recorded transaction. Therefore, the auditor would need to link every address to transacting parties. This could be done if participants provide the auditor with addresses used in each transaction. However, the auditor may not be able to identify participants who fail to report their addresses. Thus, accessibility cannot be ensured. Chapter 5 and the Annex provide details of experiments on auditing scenarios where one-time addresses are generated using HD wallet.[40]

4.2.8 Mixing

When a mixing service (centralised or P2P) is used, the ledger only records mixed and aggregated transactions. In the case of centralised mixing service, the mixing entity would possess all the original transaction information submitted to it from participants. The auditor could treat this entity as a trusted source, which would provide the auditor with the information linking individual senders and receivers. In this case, accessibility and reliability could be ensured. In addition, since this process does not require information sharing from participants, a sufficient level of efficiency could be ensured.

When mixing is P2P-based, third parties, including the auditor, cannot determine the transacting relationship between parties from the information recorded on the ledger. Thus, while the auditor could request each identifiable participant to submit the original information on its transacting relationship, it cannot be certain that the obtained information is indeed correct. Hence, accessibility could be ensured, but not reliability.

4.2.9 Ring signature

Ring signature ensures confidentiality by dissociating the transactional relationship between the sender and the receiver. Despite being able to view the transaction information written on the ledger, the auditor would not be able to single out the sender among all plausible signers (i.e. ring members). Since the sender could

---

39 In cases where ZKP is used to hide only the transaction amount, the assessment result would be similar to that of Pedersen commitment.

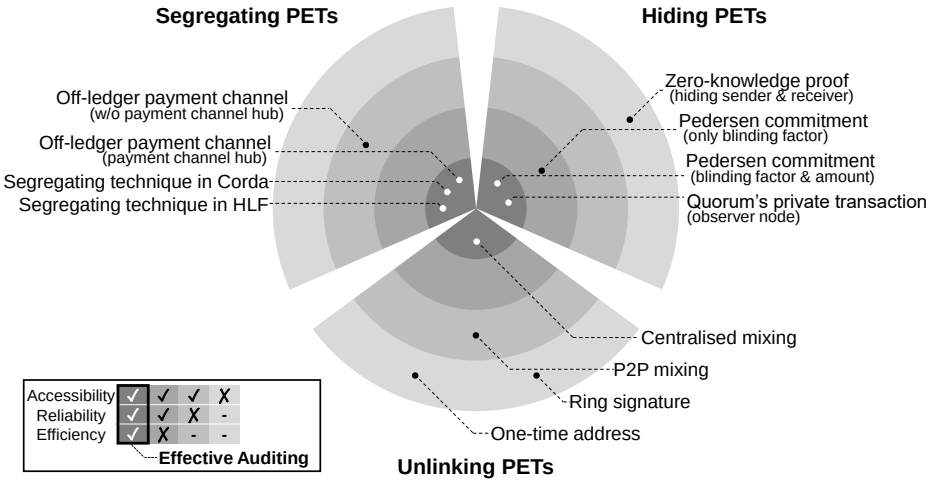
40 There are techniques attempting to link different one-time addresses that likely belong to the same participant (address clustering) and then to link at least one of these addresses to a particular participant (address tagging). However, the use of such techniques for auditing has been left out of the scope of this report. For more details on address clustering and address tagging, see *BitIodine: extracting intelligence from the Bitcoin network*, M. Spagnuolo, F. Maggi and S. Zanero, March 2014.

select public keys of all ring members without their consent, there is no evidence available for third parties to single out the sender. Therefore, accessibility cannot be ensured when ring signature is used.

Assessments of selected auditing arrangements for each PET from the proposed perspectives is summarised in Figure 8.

Figure 8

Assessment results for selected auditing arrangements



4.3 Further consideration for practical application

This section raises points to be considered further for practical application.

4.3.1 Reliance on a trusted source

The assessment in this chapter has shown that several PET setups that enable effective auditing rely on an existing, central trusted source to share the necessary information for auditing. While reliance on an existing central DLT component or a credible third party as the source of information has obvious benefits for auditing from all three perspectives, it may become a single point of failure in the auditing arrangement. Moreover, as it also becomes a single point of failure for the network, the benefits of a distributed system (such as robustness and availability) may be undermined.

If there is a single entity enabling the application of PETs, its failure would impair the provision of confidential transactions. In addition, if there is a single entity storing all transaction information, a potential security breach could result in an exposure of the transactional details of all participants. In cases where the single entity is an essential DLT component, its failure may compromise the functioning of the entire DLT network.

4.3.2 Combinations of PETs

As mentioned in Chapter 3, there are cases where multiple PETs are used complementarily to enhance the level of confidentiality of transactions. On the other hand, there could be a trade-off between enhancing confidentiality and effective auditability when PETs are applied in combination. The auditing process may become less efficient, while accessibility and reliability could be impaired. An illustrative example where Pedersen commitment and HD wallet are combined is briefly considered below.

Pedersen commitment hides the transaction amount while HD wallet unlinks sender and receiver identities from the pseudonyms in the recorded transaction information. The use of these PETs in combination could enhance the confidentiality of transaction information. While the auditor would be able to identify transacting parties when only Pedersen commitment is used, combining this with HD wallet introduces the possibility that the auditor cannot identify the transacting parties when they fail to cooperate. Therefore, accessibility cannot be ensured with this combination.

4.3.3 Multiple systems, multiple tiers and end-users

This report assumes a simplified model where transactions are conducted in a single DLT network. For practical application, the model would need to be extended to accommodate multiple payment and settlement systems as well as multi-tiered payment systems. These extensions may pose additional challenges, especially when these systems have different requirements for confidentiality and auditability. It would be necessary to coordinate different standards and processes between systems while maintaining the balance between confidentiality and auditability within each system.

This report focuses on back-end arrangements and does not consider end-users. The inclusion of end-users may increase the complexity of managing the confidentiality of end-user information recorded on the ledger and necessitate the creation of appropriate standards to determine the transactions to be audited.

4.4 Summary

The auditability of transactions in a DLT system on which PETs are applied can be assessed from the following perspectives: accessibility to necessary information, the reliability of the obtained information and the efficiency of the auditing process. For effective auditing to take place, DLT systems would need to demonstrate sufficiently positive assessment results from each of these perspectives. These perspectives could be referenced in discussions on the designs of DLT-based systems, striking a balance between confidentiality and the auditability of transaction information.

By assessing potential auditing arrangements from these perspectives, one would find different levels of auditability depending on the implementation of PETs. On the one hand, there are arrangements that do not accommodate auditing, and on the other hand, those that accommodate effective auditing. Effective auditing is enabled in auditing arrangements where (i) the auditor obtains the necessary information from trusted sources or (ii) the auditor obtains the necessary information from identifiable participants and has the means of verifying the correctness of the obtained information using information recorded on the ledger, and the entire process could be conducted without consuming excessive resources.

Some general features of PET categories with regards to auditability could be deduced from the assessment. For Segregating PETs, there is no shared ledger with records of all transaction information which the auditor can use to validate whether participants are submitting the correct information. In this sense, effective auditing requires the auditor to have access to all subsets of the ledger or to rely on a source that has records of all transactions. For Hiding PETs, the hidden transaction information is recorded on the shared ledger in verifiable form. Therefore, the key to achieving effective auditability is ensuring accessibility. For Unlinking PETs, the primary feature of these PETs is to make it difficult to determine transacting relationships from the information recorded on the shared ledger. Therefore, a mechanism to store the original set of information on sender/receiver identities and their transacting relationship and to share them with the auditor is a prerequisite for achieving effective auditability.

This chapter also raised points to be considered further for practical application. The presence of a trusted source could pose single point of failure risks to the network. There may be trade-offs between using multiple PETs in combination to enhance confidentiality and ensuring effective auditability. Extending the model to accommodate multiple payment and settlement systems, multi-tiered payment systems and end-users could pose additional challenges.

5 Experiments on selected PETs

This chapter explains the working principles and presents technical descriptions of Pedersen commitment and HD wallet and describes the experiments conducted from the viewpoint of auditability. These PETs represent the basic concepts of hiding and unlinking and are widely used in various schemes and projects. Furthermore, as discussed in Chapter 4, effective auditing could be conducted with transactions using Pedersen commitment, while accessibility to necessary information cannot be ensured for transactions using HD wallet. Stella phase 4 designed and conducted experiments on these PETs in order to consider whether effective auditing processes can be achieved on the DLT network that equips them. The following sections provide theoretical descriptions and experimental results for the two PETs.

5.1 Pedersen commitment

Pedersen commitment is categorised as a hiding PET and makes the transaction amount confidential, as explained in Chapter 3. It could be used in a range of applications for payments and settlements in DLT systems. For the purpose of the experiments described, a UTXO-based system[41] is assumed.

5.1.1 Technical description

The main concept of Pedersen commitment is creating commitments from amounts so that the sum of the commitments of the amounts equals the commitment of the sum of amounts. That is, for given two values v_1 and v_2, commitments C satisfy $C(v_1) + C(v_2) = C(v_1 + v_2)$.

Based on this idea, if a series of amounts $\{v_i\}$ fulfils $v_1 + v_2 = v_3 + v_4$, then the corresponding commitments $\{C_i\}$ can be created such that $C_1 + C_2 = C_3 + C_4$ (for simplicity, hereafter assume $1 \leq i \leq 4$). Based on this property, by replacing the transaction amounts with the corresponding commitments, a transaction sender can create confidential transaction information that can be verified by any participant without revealing the exact amounts.

In common implementations, a commitment is created based on elliptic curve cryptography and uses four different parameters. Two of them, G and H, are usually selected at the network level during setup. G is normally the generator of the elliptic curve and H represents another generator of the same elliptic curve. These generators are known to all members of the network to ensure that verification can be done. Another parameter is the blinding factor, bf, which is selected by the sender. The last parameter, v, is the amount which is meant to be hidden. The commitment is calculated as follows:

$$C = bf \cdot G + v \cdot H$$

Pedersen commitment has the following features. Stella phase 4 analysed these features through experiments.

- By appropriately selecting blinding factors[42], series of commitments $\{C_i\}$ can be created from $\{v_i\}$, and a third party can verify whether the statement $v_1 + v_2 = v_3 + v_4$ is fulfilled using the commitments instead of the amounts.

---

41 Unspent transaction output (UTXO) is a format to describe amounts in transactions. Each transaction has one or more input amounts and one or more output amounts, and transaction verification is performed based on checking sum of inputs and sum of outputs.

42 To ensure that the sum of the commitments equals the commitment of the sums, the blinding factors also need to be calculated in order that they satisfy the same condition that the amount is fulfilling, which in this case is $bf_1 + bf_2 = bf_3 + bf_4$.

- If $v_1 + v_2 \neq v_3 + v_4$, creating $\{C_i\}$ such that $C_1 + C_2 = C_3 + C_4$ is computationally infeasible. Thus, it is infeasible for participants to include incorrect amounts in a transaction.

- Even if $\{v_i\}$ includes negative value(s), so long as $v_1 + v_2 = v_3 + v_4$ holds, verifiable $\{C_i\}$ can be created, where the negative value is completely hidden. Thus, participants may potentially create a larger output than the sum of inputs by including negative outputs.[43]

- There is a "discrete logarithm" x with $H = x \cdot G$; however, it is computationally infeasible to find x. Nevertheless, if a participant obtains x, it can be used to break the binding property of Pedersen commitment and can change the committed amount.[44]

5.1.2 Auditability of Pedersen commitment

The auditability of Pedersen commitment is defined as the auditor's ability to interpret the commitments and verify the hidden transaction amounts. The experiments were designed to analyse possible ways in which the auditor could audit the amounts hidden in the commitments. To interpret the commitment, the auditor would need the information on the blinding factor and/or the hidden amount. Stella phase 4 defined three scenarios on the types of information shared with the auditor and ran experiments to analyse the viability of auditing.

If the auditor receives both the blinding factor and the hidden amount, it can always know if the information received is correct. If the information is valid, auditing is possible.

If the auditor receives the public key of the blinding factor ($bf \cdot G$) and the hidden amount, it also needs to receive the signature made using bf from the sender of the information to check the validity of the amount.[45] If the information is valid, auditing is always possible.

If the information the auditor receives lacks the hidden amount and only receives (i) the blinding factor or (ii) public key of the blinding factor and a signature, the auditor cannot immediately find the exact amount in the commitment. Although the auditor can guess the amount by brute force, the calculation may take a significant amount of time if the number of potential amounts is not limited.

As discussed in Chapter 4, it could be assumed that accessibility and reliability are ensured for the three scenarios contemplated in the experiments. However, there are

---

43 To prevent participants' misbehaviour, such as including negative values in a transaction, a range proof mechanism is combined with Pedersen commitment in practice.

44 Practically, parameters H and G are determined in a way that participants can be sure that the discrete logarithm assumption cannot be broken by any party. A common way of ensuring this is by being transparent on the process followed to calculate the parameters. Otherwise, participants could suspect that the party determining the parameters knows the relationship x.

45 The public key ($bf \cdot G$) itself can be calculated by anyone given C and any v (because $bf \cdot G = C - v \cdot H$). Therefore, the auditor needs to confirm whether the participant knows bf.

differences in the efficiency of the auditing process among these scenarios, which could be considered based on the computational burden of the auditor. If the auditor receives (i) both the blinding factor and the amount or (ii) the public key of the blinding factor, its signature and the amount, the computation burden is minimal and one could consider the process sufficiently efficient. On the other hand, if the auditor only receives the blinding factor and the number of potential amounts is not limited, more computational power would be needed and it is expected that the process would not be sufficiently efficient.

5.2 Hierarchical deterministic wallet

HD wallet provides a well-manageable scheme for one-time address derivation and is categorised as an Unlinking PET. The main underlying concept is the key derivation function that derives numerous addresses from one secret value. HD wallet is proposed and defined in the Bitcoin community[46] and implemented in various wallet applications.

5.2.1 Technical description

In HD wallet, all keys are derived from a seed (Figure 9). With the adequate transformations, this seed is used to derive the master private key of the wallet. This master key is the genesis of the wallet: the rest of the keys can be created from this starting point. The master key is used to derive numerous keys, and each of these keys is used to derive numerous other keys.[47] More generally, any key in HD wallet can be regarded as a parent key and derive numerous child keys. Therefore, HD wallet can hold a large number of keys in a tree-format, and its owner only has to manage the seed and derivation path[48] for each key.

There are two methods for deriving keys: hardened derivation and non-hardened derivation. Non-hardened derivation provides both private key and public key derivation schemes. With respect to non-hardened public key derivation, child public keys are generated from their extended parent public key without using the private key. Therefore, this method is preferred when the owner needs to share a subset of addresses with third parties without revealing any private keys. The hardened derivation is used to derive private keys in a more secure manner. The auditing process considered in this report is based on the non-hardened public key derivation.

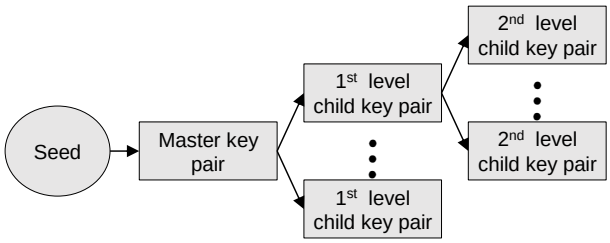
---

46 See BIP-0032 https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki and BIP-0044 https://github.com/bitcoin/bips/blob/master/bip-0044.mediawiki.

47 The key derivation is enabled by using the parent key itself and additional information. The key which has this additional information is called the “extended key”.

48 Derivation path is an index of the address in the derivation tree. For example, “m/1/2/3” indicates the third child private key of the second child of the first child of master key. The path of a private key starts with “m” and that of a public key with “M”. If the path includes an apostrophe (’), it means that it is derived with hardened derivation, while if does not, it is derived with non-hardened derivation.

Figure 9
Hierarchical key derivation in HD wallet



Stella phase 4 analysed the following features of HD wallet through experiments:

- Generating a seed from a set of words[49] and transforming the seed into the master private key of a wallet.
- Implementing three functions to derive keys: the hardened key derivation, the non-hardened private key derivation and the non-hardened public key derivation.
- Confirming the weakness of non-hardened derivation, that is, if a party has information about an extended public key and any of the child private keys of that branch is leaked, it becomes possible for the party to derive the parent private key, exposing a whole branch stemming from the parent private key. This is not the case for hardened key derivation.

5.2.2 Auditability of hierarchical deterministic wallet

As described in Chapter 4, effective auditing cannot be achieved with HD wallet since accessibility to necessary information cannot be ensured. These affirmations are based on the experiments on HD wallets, which focus on whether the auditor could correctly identify all the pseudonyms (public keys/addresses) that a particular participant has used in transactions. Since private keys should not be shared with third parties, Stella phase 4 conducts experiments based on the non-hardened public key derivation. This allows for the derivation of child public keys from their parent extended public key. If the auditor has the extended public key of the participant, it could use the extended public key to derive child public keys and audit the transactions in which these keys were used.

To analyse auditability of HD wallet from accessibility perspective, Stella phase 4 carried out simulations of the auditing process as described below.

- Preparing a situation where a participant derived numerous addresses in its HD wallet and used them in each transaction. In this experiment, sets of two hundred addresses were generated according to certain different patterns.

[49] Based on BIP-0039 https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki

- If the extended public key and the derivation path are shared with the auditor, the auditor can restore all addresses and conduct effective auditing for each transaction.
- If only the extended public key of these addresses (and not the derivation path) is shared with the auditor, the auditor has to generate all possible addresses (about two billion addresses are derived from a parent extended public key) and collate them with addresses used in each transaction.[50] The experiment suggested that in this case, the calculation could take a long time and consume computational power to the extent that the auditing may be impractical.
- If the addresses in a transaction are not derived from the extended public key shared with the auditor, then the auditor does not have viable means to identify the transacting parties.

From these experimental results, it is clear that when participants fail to cooperate with the auditor (e.g. they for some reason use addresses that are not derived from the extended public key shared with the auditor or do not communicate the path), then accessibility to necessary information cannot be ensured. This is because the auditor cannot generate with certainty all addresses that a particular participant used. For effective auditing to take place, there needs to be a mechanism which ensures that the participants always derive and use the keys in a way that the auditor can identify the transacting parties.

[50] Under specific circumstances, if the auditor is able to find a pattern in the way the participant derives addresses, this process could be feasible.

Annex

This annex introduces a more detailed description of the working principles of Pedersen commitment and HD wallet and the experiments conducted. Its purpose, therefore, is to complement the description in Chapter 5 with additional details, including a more comprehensive but informal description of the mathematics behind these techniques.

A.1 Pedersen commitment

Cryptographic techniques have been developed that allow encrypting transaction amounts in a way that nodes could verify transactions without decrypting them. One technique that can be used for that purpose is Pedersen commitment, which is analysed in more detail in the following section. The experiments described were developed using Python.[51]

A.1.1 Basic features

In Pedersen commitment, a commitment is created to encrypt a particular piece of information:

$$C = com(blinding\ factor \| information)$$

This is no different to other commitment techniques like, for example, SHA. However, with Pedersen commitment, it is true that:

$$com(bf_1 \| info_1) + com(bf_2 \| info_2) = com(bf_1 + bf_2 \| info_1 + info_2)$$

This means that the sum of the commitments is equal to the commitment of the sums. To create a Pedersen commitment using elliptic curves, the first step is to select a curve. A commitment can be understood as some sort of public key of the amount to commit. This means that the generator (G) of the curve needs to be used to generate the commitments.

However, if only amounts and G are used, for low-value amounts a brute-force approach could easily break the encryption. To avoid this, another point in the elliptic curve is generated, H. Therefore, the commitment is calculated as follows:

$$C = bf \cdot G + v \cdot H$$

This has an additional feature: it makes it impossible to decrypt the commitment without information about the blinding factor and the amount itself, as there is a large number of blinding factor and amount pairs that could give the same C. This is why Pedersen commitment is generally described as perfectly hiding.

Although there is a large number of pairs of blinding factor and amount that lead to the same commitment, it is very difficult to find two such pairs. This is commonly

[51] The library used for the experimentation with Pedersen commitment was fastecdsa (https://pypi.org/project/fastecdsa/).

known as computationally binding: unless a party has enough computing power to find two pairs that yield the same commitment, the transaction amount hidden in a commitment cannot be changed.

In the developed implementation, some of these characteristics were analysed. For simplicity, all verifications were performed in the elliptic curve space (all commitments are points on the elliptic curve).

The first experiment addressed the creation of a set of commitments and the verification that the sum of the commitments of the amounts equals the commitment of the sum of the amounts. A set of commitments was created for a particular sequence of amounts (A+B = C+D) and the blinding factors were adjusted to also comply with the equality condition.

Similarly, the second experiment addressed the inequality condition: if the discrete logarithm is not broken, different amounts cannot give the same commitment. Blinding factors and amount are chosen so that A+B is not equal to C+D and therefore the commitments are also different.

A.1.2 Potential weaknesses

The elliptic curve group used for elliptic cryptography is cyclic. One consequence of this is that the addition of large numbers can result in an “overflow” and its behaviour is no different from that of a negative value.

This makes the statement “the sum of the commitments is equal to the commitment of the sums” also applicable to sums which involve negative values. And since transaction verification is made using the commitments, instead of transaction amounts, those nodes may not notice that negative values have been introduced. One side effect of this could be the creation of new assets.

This was illustrated in the implementation with an example. The goal was to create a set of values that, de facto, generated an additional 85 units (10 + 5 = 100 - 85). Since both sides of the equation add up to the same value (15), the commitment of the sum of each side is the same and, therefore, such a transaction may be considered valid. A common approach to address this issue is to provide also range proofs, or proofs that the amounts lie within certain ranges.

Another potential weakness comes from the discrete logarithm assumption regarding the points that are used to generate the commitments. Since both H and G are points in the same elliptic curve, it is true that there could always be a value x so that $H = x \cdot G$.

Following the discrete logarithm assumption, it is computationally infeasible to calculate x. However, there could be a way around the discrete logarithm assumption. To ensure that the commitments can be verified by any node in the network, H and G have to be settled at the network level. In the implementation, it is assumed that the party that generates H makes it in a way that the relationship between H and G is known to this party. If there is a commitment:

$$C = bf_1 \cdot G + v_1 \cdot H$$

There is at least another pair for which the following statement is true:

$$C = bf_1 \cdot G + v_1 \cdot H = bf_2 \cdot G + v_2 \cdot H$$

Since a potentially misbehaving party may have an interest in claiming a value larger than the original commitment (creating money), it is assumed that v_2 is larger than v_1. How can bf_2 be calculated?

$$bf_1 \cdot G + v_1 \cdot H = bf_2 \cdot G + v_2 \cdot H$$

$$bf_2 \cdot G = bf_1 \cdot G + (v_1 - v_2) \cdot H$$

In principle, the relationship between G and H is not known, so calculating bf_2 should not be computationally feasible due to the discrete logarithm assumption. However, if the party that has generated H knows this relationship (x), it can use it to calculate bf_2:

$$bf_2 \cdot G = bf_1 \cdot G + (v_1 - v_2) \cdot x \cdot G$$

$$bf_2 = bf_1 + (v_1 - v_2) \cdot x$$

In the implementation, this formulation was verified and the same commitment was obtained taking advantage of knowing the relationship between H and G. This demonstrates potential consequences if the discrete logarithm is broken. Breaking the discrete logarithm assumption, however, will require a disruptive change in technology (e.g. quantum computing). It also shows that a system which uses Pedersen commitment to hide transaction amounts must ensure that H is generated in a way that no party has knowledge about the relationship between G and H to prevent them from using that knowledge to alter the amounts hidden in the commitments.

A.1.3 Auditability and conclusions

The auditability of Pedersen commitment is understood as the auditor's ability to open the commitments and verify the hidden transaction amounts.

As previously mentioned, the commitment scheme is described as perfectly hiding. This means that the auditor needs to have some information to be able to open the commitment. Three different approaches are analysed in which the auditor could verify the reliability of the information received. The underlying assumption is that the discrete logarithm assumption is always valid.

Iteration 1: Participants share all the details of the commitments with the auditor when requested. In this iteration, the auditor is always able to open the commitment as long as the participant provides it with the blinding factor and the transaction amount.

Iteration 2: Participants share the value of the transaction with the auditor but instead of also sharing the blinding factor, they share the public key of the blinding factor: $Pub = bf \cdot G$

This information can then be used to open the commitment.

$$C = bf \cdot G + v \cdot H = Pub + v \cdot H$$

However, a participant could choose the public key of the blinding factor so that it fits a new value chosen by the participant:

$$C = bf_1 \cdot G + v_1 \cdot H = bf_2 \cdot G + v_2 \cdot H$$

$$Pub_1 + v_1 \cdot H = Pub_2 + v_2 \cdot H$$

$$Pub_2 = Pub_1 + (v_1 - v_2) \cdot H$$

One way in which the auditor could be certain of the validity of the blinding factor public key is using signatures of the blinding factor. If the participant signs the commitment with the blinding factor, the auditor can use that signature to verify the validity of the blinding factor public key that has been shared.

Iteration 3: Participants only share the blinding factor or the blinding factor public key (with signature) with the auditor, but not the transaction amount. This means that the auditor would need to try all possible amounts until the correct one is found.[52]

Multiple simulations of brute-force derivation with amounts of seven and six digits were conducted. While running on one single core, the results show that a significant amount of time would be consumed. However, this process can be accelerated with parallel computing, which suggests that depending on the computing capabilities of the auditor, transacting amounts could be limited to a certain range so that participants only have to share the blinding factor or its public key.

It can be concluded that the auditor can verify the validity of the information shared and, if this information is valid, conduct a successful audit. If the auditor receives both the blinding factor and the amount or the blinding factor private key and the amount, the computation burden is minimal. In this scenario, it could be considered that the process is sufficiently efficient. On the other hand, if the auditor only receives the blinding factor or the blinding factor public key and the range of amounts is not limited, more computational power would be needed and it could be expected that the process would not be efficient.

---

[52] If the participant shares a blinding factor other than the one used to create the commitment, the auditor will not be able to find an amount, since for the participant to share a blinding factor that gives a valid amount, the discrete logarithm assumption would need to have been broken.

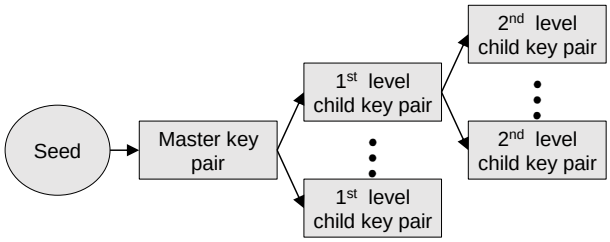
A.2 Hierarchical deterministic wallet

HD wallet is a widely used PET in several DLT networks, such as Bitcoin or Ethereum. The rules followed in the implementation to generate keys in HD wallet are described in BIP-0032.[53] In the following section, an introduction to HD wallet is given, as well as a description of the experiments conducted. The experiments were developed using Python[54] and the results obtained were validated against known implementations of HD wallet.

A.2.1 Basic features

With HD wallet, all pseudonyms are derived from a starting point called the seed. The common term for the pseudonym is public key, and the common way to derive each public key is from its private key using elliptic curve cryptography. From the seed, a master key pair is generated[55], from which all subsequent key pairs are generated in a tree-like structure as shown in Figure A.

Figure A
Hierarchical key derivation in HD wallet



From seed to master private key

The master private key is generated from a seed that contains randomly generated bits. According to BIP-0032, the length of the seed has to be between 128 and 512 bits.

The first function developed for the experiments is the generation of the master private key from the seed. This function only requires calculating the HMAC[56]-SHA512 using “Bitcoin seed” as the key and the seed as the message to calculate the HMAC. A diagram of this process can be seen on Figure B.

53 https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki

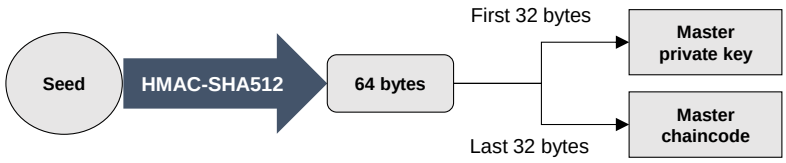
54 The libraries used in the experimentation: fastecdsa (https://pypi.org/project/fastecdsa/), hmac (https://docs.python.org/3/library/hmac.html), base58 (https://github.com/keis/base58), bitcoinlib (https://pypi.org/project/bitcoinlib/).

55 A key pair consists on a pair of private and public keys.

56 HMAC, hash-based message authentication code, is a type of message authentication code that involves a cryptographic hash function and a secret key.

Figure B

Master private key calculation from seed



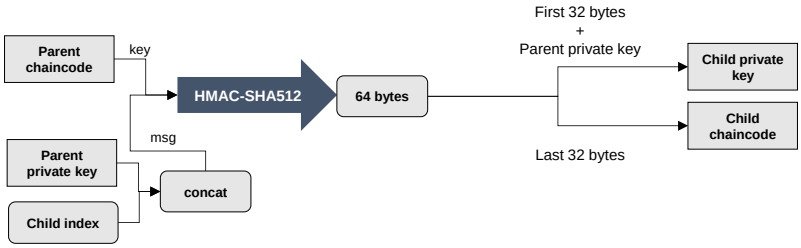
This gives 512 bits, which are 64 bytes.[57] The first 32 bytes will be used to calculate the master private key, while the last 32 bytes are its chaincode. As the final step of the process, it needs to be checked that the first 32 bytes interpreted as a number have a lower value than the order of the curve.[58] This value is the master private key.

Hardened derivation

Hardened derivation is a technique to generate a child private key from a particular private key. The methodology followed in the key generation process is illustrated in Figure C.

Figure C

Hardened key derivation



As illustrated in Figure C, three pieces of information are required in order to apply hardened derivation: the private key[59], the chaincode of the key and the index of the child. Since hardened derivation is being used, the index has to be an integer

[57] One byte is 8 bits.

[58] This check is common in elliptic curve cryptography and is related to finite fields. More on this can be found at https://andrea.corbellini.name/2015/05/23/elliptic-curve-cryptography-finite-fields-and-discrete-logarithms/.

[59] A private key is 32 bytes long. The process requires including 0x00 before the 32 bytes of the keys, so it is 33 bytes long.

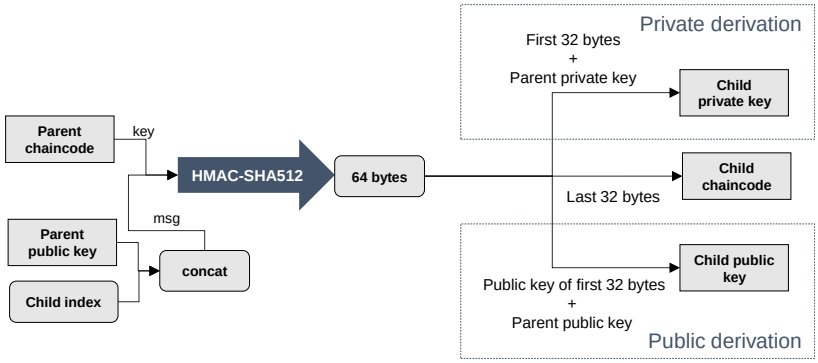
between 2^{31} and $2^{32} - 1$. The private key is concatenated with the index and the HMAC-SHA512 is calculated with the chaincode as key.

The result obtained is 64 bytes long. The last 32 bytes are the chaincode of the child private key, while the first 32 bytes are added to the parent's private key to become the child private key.[60] A function to generate private keys using hardened derivation was developed. To verify the results, it was used to generate account level private keys. To do so, hardened derivation was applied three times, being the first one on the master private key.

Non-hardened derivation

Non-hardened derivation is a technique for child key derivation that also allows generating child public keys from parent public keys. To ensure this feature, the child private key generation of the non-hardened derivation is different from the hardened derivation. Therefore, there are two non-hardened generation techniques: the private derivation (child private key) and the public derivation (child public key). A diagram of both of these derivation processes is shown in Figure D.

Figure D
Non-hardened child key derivation



Both processes start using the public key of the parent, its chaincode and the index of the desired child key. Similar to the hardened derivation, the public key and the index are concatenated and the HMAC-SHA512 is calculated with the chaincode as the key. The result is a string of 64 bytes, from which the last 32 bytes are the chaincode of the child key. Up to this point, non-hardened derivation is the same for private and public derivation. However, the processes slightly differentiate as follows:

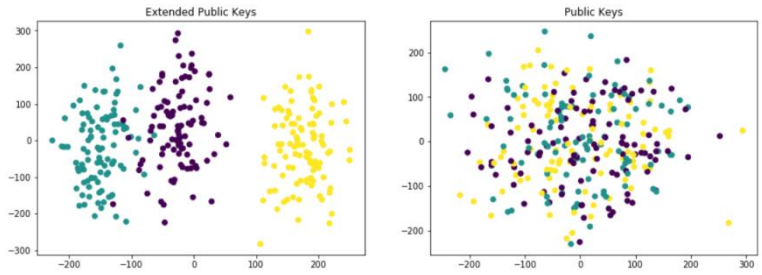
[60] Again, this 32 bytes are compared to the order of the curve and the modulo operation is applied to ensure that this value is smaller than the order of the curve.

- In private derivation, add the first 32 bytes to the parent private key to generate the child private key.[61]
- In public derivation, calculate the public key of the first 32 bytes and add it to the parent's public key to obtain the child's public key.[62]

Extended and non-extended keys

So far, the different techniques that could be used to derive child keys in HD wallet have been described. To demonstrate other aspects of keys, Figure E shows a visualisation of 300 public keys from three different wallets (100 from each wallet). The left graph represents extended public keys, while the right graph shows normal public keys.

Figure E
Representation of extended public keys and public keys



The public keys of the three wallets do not exhibit common patterns. On the other hand, the extended versions of those public keys show a clear pattern: it can be seen that the extended version of the public key includes some information that allows identifying related public keys. The extended version of any key (regardless whether public or private) has the size of 78 bytes, and each of them has a specific meaning.

- **vbytes:** There are four bytes used to specify the version. It is the same for all public keys on the same network.
- **Depth:** One byte value that represents the depth of the key in the hierarchical wallet.
- **Fingerprint:** It is a four-byte value that identifies the parent. For the master public key it is a four-byte zero, but for the rest of the keys it is the hash160 of the parent.
- **Index:** Four bytes represent the index of the child.

61 Modulo operation is applied to ensure it is lower than the order of the curve.

62 What should be understood here is adding the points in the elliptic curve that correspond to those public keys.

- **Chaincode:** Chaincode of the key.
- **Key:** 33 bytes represent the key.

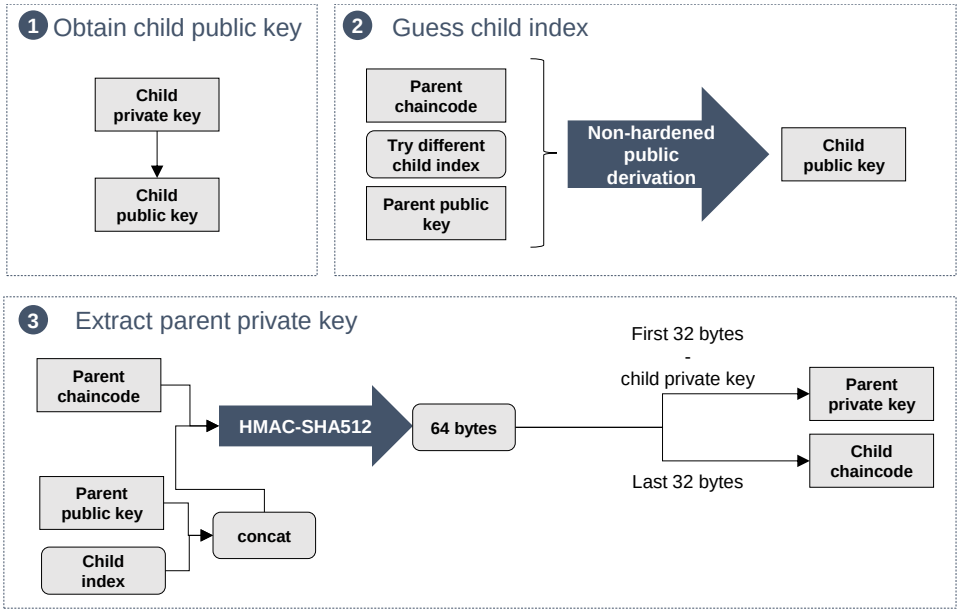
Therefore, it could be hinted that since the fingerprint is a key identifier, the extended keys in Figure E appear to form several clusters because of the same fingerprint. In particular, it was verified that, as expected, the fingerprint of the keys of the same colour has the same value, as they have a common parent.

A.2.2 Potential weaknesses

The technique that allows deriving public keys from other public keys is one of the main causes of the weaknesses of HD wallet. If a single child private key is leaked and the extended parent public key is known, the parent private key and all of the other child private keys are compromised.

Figure F shows how this weakness could potentially be exploited.

Figure F
Obtaining a parent private key from a child private key



First, the parent key pair is generated. From the key pair, a child private key is generated. It is assumed that the index is unknown, so the first step is to use the child private key to generate a child public key. Using public key derivation, the index of the key can be found, which then can be used to calculate the parent private key.

This, however, is not possible with hardened derivation because the parent private key is needed for the first step of the of child key derivation (while with the non-hardened derivation the first step requires the public key). Therefore, while hardened derivation does not offer the possibility of deriving child public keys from the parent public key, the exposure of one child private key only affects that particular child key and not the parent or the rest of the child keys.

A.2.3 Auditability and conclusions

Non-hardened derivation allows for the derivation of child public keys from the account public key. This means that if the account public key is shared with the auditor, all the child public keys could be generated without any knowledge of the private keys that enables the funds to be spent. Therefore, non-hardened derivation seems more suitable for auditing, as opposed to hardened derivation.

The experiments are defined as a simulation of what an auditor could encounter to analyse what aspects should be taken into consideration when auditing. It is assumed that the auditing party only has information about the public key of the account in which the participant is generating keys. Different scenarios have been analysed, in which participants have generated 200 keys following different patterns that are unknown to the auditor.

Iteration 1: Participants start generating keys in ascending order, starting from index 0. In this iteration, the auditor searches for public keys in ascending order, one by one. In this case, the auditor is able to easily identify all the participant's public keys relatively quickly. However, this only occurs since the search pattern of the auditor and the generation pattern of the user were aligned. Similarly, this situation could become more time-demanding and computationally heavier with a higher number of public keys to search.

Iteration 2: The participant generates keys one by one in descending order, starting from the highest index. If the auditor uses the strategy of the previous iteration, it fails to recover any key within a reasonable time. Therefore, it adapts and changes the search pattern to descending search. With the new search pattern, all keys generated by the user are found with ease. Similar to the previous iteration, this could become more time-demanding with a high number of keys to search for.

Iteration 3: The participant generates keys using only indexes according to a polynomial function of order three.[63] If the auditor uses a descending search, it does not find any key within a reasonable time. If the auditor uses an ascending search, it is only able to find 19 public keys within a reasonable time. With the public keys that the auditor has generated, it could try to guess the pattern followed by the participant. There are several regression or machine learning techniques that could be employed trying to find the pattern for key generation that the participant follows. In the conducted experiment, the chosen approach was the attempt to fit a polynomial regression. By fitting the points to a curve, the auditor is able to find a pattern in the way the user generates keys. Having been able to guess the pattern, the auditor is able to find all of the participant's keys within a very short time. Therefore, similar to the previous two iterations, when the auditor knows the pattern for key generation used by the participant, the auditor can generate the public keys with relative ease.

---

[63] A polynomial function of order 3 has the following form: $f(x) = a + b \cdot x + c \cdot x^2 + d \cdot x^3$

Iteration 4: Nevertheless, a participant could change its pattern, and, in an extreme case, use no pattern at all. In the fourth iteration, the participant does not follow any pattern that could be guessed by the auditor. The participant randomly chooses the index each time a new key is created and keeps a list of the already used indexes to avoid using the key twice. In this iteration, the auditor applies both the ascending and descending search strategies, recovering 21 keys in total. However, no pattern can be found from the indexes (as there is none) and, therefore, the only way through is to generate all the possible keys (2^{31}), which does not seem like an efficient strategy. On the other hand, if the participant shares its list of used indexes (this would represent the pattern followed by the participant) with the auditor, the keys are recovered very easily.

The experiments suggest that auditing is possible as long as participants cooperate with the auditor. Here, cooperation refers to sharing the account public key and the pattern used for key derivation. However, participants could misbehave in two different ways. The first one would be to share both pieces of information but use a different pattern than the one communicated to the auditor. The second way would be to use a different account key than the one shared with the auditor, and thus completely hide a branch.[64] Therefore, the auditor cannot be sure it has the required information related to a particular participant and a successful audit could not be guaranteed.

[64] This affirmation was also tested with a subset of the keys. The reason for this to happen is that the collision (the probability of generating the same address) between wallets and branches within a wallet is extremely low (as SHA512 is involved in key derivation, both from seed and parent key).

Postal address 60640 Frankfurt am Main, Germany
Telephone +49 69 1344 0
Website www.ecb.europa.eu

Postal address 2-1-1 Nihonbashi-Hongokucho, Chuo-ku, Tokyo 103-0021, Japan
Telephone +81 3 3279 1111
Website www.boj.or.jp