Securing Wireless Payment Channel Networks With Minimum Lock Time Windows

Gabriel Antonio Fontes Rebello, Maria Potop-Butucaru, Marcelo Dias de Amorim, Otto Carlos Muniz Bandeira Duarte

Securing Wireless Payment Channel Networks With Minimum Lock Time Windows*

Gabriel Antonio F. Rebello[1,2], Maria Potop-Butucaru[2],
Marcelo Dias de Amorim[2], and Otto Carlos M. B. Duarte[1]

[1]Universidade Federal do Rio de Janeiro, Brazil [2]LIP6/CNRS - Sorbonne Université, France

Abstract

Payment channel networks (PCN) enhance the impact of cryptocurrencies by providing a fast and consensus-free solution to the scalability problems of traditional blockchain protocols. However, PCNs often rely on powerful nodes with high availability and computational capacity, hindering their adoption in mobile environments. In this paper, we consider a hybrid PCN architecture that extends the functionalities of traditional PCNs to wireless resource-constrained devices. We analyze the token theft vulnerability and propose a countermeasure based on lock time windows. We evaluate our proposal with real data from the Lightning Network and from mobile broadband networks. The results show that the minimum lock time window depends on the downtime of devices and that selecting a default window is most effective when devices present high availability.

Keywords: blockchain, payment channel networks, wireless

ACM Reference Format:
Gabriel Antonio F. Rebello[1,2], Maria Potop-Butucaru[2],, Marcelo Dias de Amorim[2], and Otto Carlos M. B. Duarte[1]. 2021. Securing

Wireless Payment Channel Networks With Minimum Lock Time Windows. In *Proceedings of ACM Conference (Conference'21).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnn.nnnnn

1 Introduction

The most used blockchain consensus protocols today present significant latency and energy expenditure issues that hinder the adoption of cryptocurrencies in everyday life [1, 2]. Publishing a transaction in Bitcoin takes approximately one hour, incurs $20 fees, and spends an amount of energy enough to maintain a United States household for one month [3]. Payment channel networks (PCN) offer a scalable and efficient off-chain solution to improve such performance by allowing transactions to occur without the need for consensus. However, despite providing an efficient solution compared to transacting directly in the blockchain, payment channel networks still rely on processing power, storage capacity, and high node availability. Nodes in Bitcoin's Lightning Network use onion routing [4, 5], while Ethereum's Raiden Network routing relies on a synchronized global topology view to define paths [6]. Other PCNs also rely on constant blockchain verifications and block storage that overload end-hosts [2, 7].

PCNs are challenging to implement in wireless devices with few resources and intermittent connectivity patterns caused by lossy wireless connections. Several works in the literature propose adaptations of the Lightning Network to mobile devices [8–10]. However, to the best of our knowledge, no solution analyzes the connectivity problem in-depth and in a PCN-agnostic manner.

Our contribution. In this paper, we make several contributions towards implementing payment channel networks in resource-constrained environments composed of wireless devices. Firstly, we consider a hybrid PCN architecture that allows devices to issue payments despite presenting intermittent connectivity and lacking the capacity to store a blockchain node. Secondly, we formulate and analyze the *token theft problem*, a vulnerability that is present in all PCNs [2, 5, 6] but becomes critical in wireless environments due to device downtime and packet loss. Thirdly, we propose a countermeasure to the token theft problem based on lock time windows that PCNs already implement. Finally, we analyze the performance of our approach using real data from the Lightning Network and mobile broadband connections.

2 Wireless Payment Channel Networks

In this section, we introduce the concepts of traditional payment channel networks and then present the network architecture we consider for wireless environments.

2.1 Payment Channel Networks (PCN)

Payment channels are blockchain-supported bidirectional connections between two parties who wish to transact with each other. Figure 1 shows an example of a payment channel. To open a payment channel, two users, Alice and Bob, sign and publish a funding transaction that transfers a fixed

*This paper was funded by CNPq, CAPES, FAPERJ and FAPESP (18/23292-0, 2015/24514-9, 2015/24485-9 2014/50937-1).

Conference'21, July 2021, City, Country

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnn.nnnnn

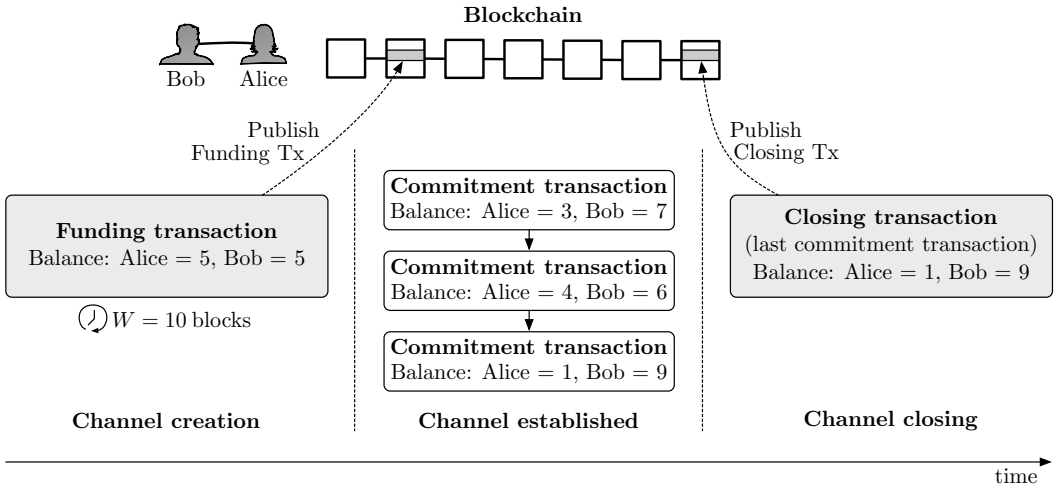

Figure 1. A payment channel between users Alice and Bob. Both users issue private commitment transactions in real time after establishing the channel. The funding transaction contains a lock time window W that locks the tokens for a predefined number of blocks once the channel is closed unilaterally.

amount of tokens to a joint address. Alice and Bob can then continuously rebalance the funds of the address by sending private signed commitment transactions. This way, Alice and Bob transact with each other without paying the blockchain fees and without waiting for the system to approve the transactions through consensus. Such an approach is especially fit for micropayments. The funding transaction contains a *lock time window*[1], W, that defines a delay, measured in blocks, that the closing party must wait to recover her/his invested tokens. To close the channel, either Alice or Bob publishes the last commitment transaction into the blockchain and waits for the lock time window to recover their tokens. The lock time window serves as a security mechanism to prevent Alice or Bob from publishing a previous balance into the blockchain. If either party detects this behavior during the lock time window, she/he can punish the other party by spending all the tokens in the channel.

Illustration. A payment channel network is a peer-to-peer (P2P) network composed of users and their payment channels. Figure 2 depicts an example of a PCN. Nodes can issue payments to destinations even if they do not share a payment channel. If Alice wants to send one token to Charlie, she can send the token to Bob, and Bob relays it to Charlie. Bob receives the token from Alice once he sends one token of his own to Charlie through Hashed Timelock Contracts (HTLC) [11], a special type of script. Hence, routing payments in a PCN is different from classical data networks because the system must ensure that every intermediary has enough funds to transfer tokens. The routing mechanism must prevent the depletion of the links in the payment path.

[1]The lock time window is often referred to as "to self delay" or "return delay" in the literature [2, 4]. We consider the terms equivalent.

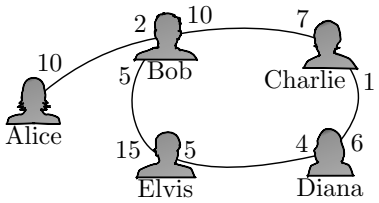

Figure 2. An example of a payment channel network (PCN) composed of bidirectional payment channels with limited capacity. User users who do not share direct links can route payments through intermediaries.

Several works in the literature explore this problem in traditional payment channel networks, such as Bitcoin's Lightning Network and Ethereum's Raiden Network [6, 12, 13].

2.2 PCNs for Wireless Resource-contrained Devices

We consider Wireless Payment Channel Networks (WPCN), a hybrid PCN architecture composed of a static and reliable core as well as peripheral unreliable mobile devices with limited resources. We argue that a hybrid topology is the most impactful because the main IoT platforms and mobile device architectures today rely on gateways and edge computing to provide services. Figure 3 depicts the topology of our network architecture. We define two types of nodes:

- **Full nodes (FN)**, which compose the core network and act as routers. Full nodes may represent service companies, telcos, or any node with computing power to store a full copy of the blockchain. Full nodes are online with high probability at all times and communicate via a reliable transport protocol. Although churn can occur in the core network, the probability that a full

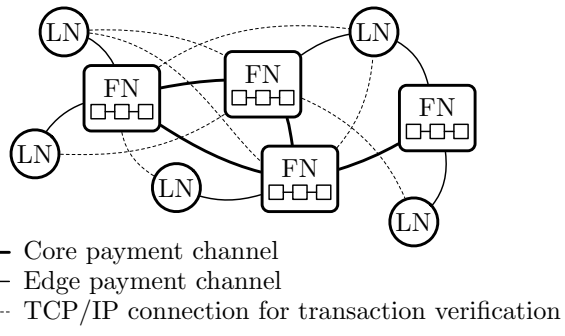

Figure 3. An example of a topology of Wireless Payment Channel Networks. The light nodes (LN) represent wireless devices, and the full nodes (FN) represent nods that store a copy of the blockchain. Light nodes establish TCP/IP connections to verify the states of their channels in the blockchain.

node quits the network without closing its payment channels is negligible compared with mobile nodes.

- **Light nodes (LN)**, which are mobile devices that connect to the network via lossy wireless connections and are resource-constrained. Light nodes may represent mobile phones, sensors, smart objects, or any IoT device that presents limited computing and storage capacities. We assume light nodes can disconnect at any time without closing the payment channel due to battery and hardware malfunction, environmental conditions, cellular operator issues, and other limitations of mobile devices. We assume light nodes establish unreliable connections to send/receive transactions and request channel state verification. They can perform public-key cryptography to sign transactions but cannot store a copy of the blockchain.

Full nodes connect to other full nodes via payment channels with a large capacity to route payments. Light nodes connect to one or more full nodes via smaller and possibly unidirectional payment channels. Henceforth we refer to channels between full nodes as *core payment channels* and channels between a light node and a full node as an *edge payment channels*. We do not consider payment channels between light nodes as it would be unlikely for two light nodes to continuously transact with each other for a long period. *Entry nodes* are the nodes a light node selects as its connections. Light nodes also establish TCP/IP connections to other full nodes to continuously verify the states of its payment channels and avoid eclipse attacks. For our formal definition of Wireless Payment Channel Networks (WPCN), we extend the definition of PCNs from Malavolta *et. al* [14]:

Definition 1 (Wireless Payment Channel Network (WPCN)). A Wireless Payment Channel Network is a time-varying directed graph $\mathbb{G}(t) := (\mathbb{V}(t), \mathbb{E}(t))$, where $\mathbb{V}(t)$ is the set of devices in the network at time t and $\mathbb{E}(t)$ is the set of payment channels that are open at time t. Any device $u \in \mathbb{G}(t)$ can alter the set $\mathbb{E}(t)$ of edges via three possible operations:

- `openChannel`$(\langle u, v\rangle, \langle \alpha, \beta\rangle, T, F, W)$ opens a payment channel $u \leftrightarrow v$ with channel identifier c_{id}, capacity (α, β), timeout T, and fee F. The lock time window W defines the lock time in which neither party can claim the tokens if the channel is closed in a unilateral manner. The operation publishes a transaction in the blockchain that must be signed by both u and v;
- `closeChannel`$(\langle u, v\rangle, Tx(t))$ closes the payment channel $u \leftrightarrow v$ with transaction $Tx(t)$, which contains the latest balance that has been signed at time t by both parties, and publishes it in the blockchain. This operation can either be issued cooperatively by having both signatures or unilaterally by either party. If the channel is closed unilaterally, the party that closed the channel can only claim her/his tokens after the predefined time window W;
- `pay`$(\langle u, v\rangle, p, V)$ transfers a value of V tokens from u to v via path $p = \langle u, r_1, r_2, ..., r_n, v\rangle$. For the scope of this paper, we assume the path p is defined by u before issuing the operation. Every hop from u to v will have its capacity decreased by V tokens in the direction of the payment receiver v if the whole path has enough capacity. Otherwise, the operation fails and all channels remain unaltered.

3 The Token Theft Problem

PCNs like the Lightning Network [5] and the Raiden Network [6] assume that any node that transacts in the network remains online while the channel is open. Otherwise, the channel counterparty may publish an old transaction to recover the tokens that she/he sent to the disconnected node. The system punishes malicious nodes by allowing the victim to spend all tokens if it recovers during the predefined lock time window. *Hence, it is only worth it to attempt the attack if the malicious node can guarantee that the other party will not verify the blockchain until the time window expires.*

A small default value for lock time windows works well for wired peer-to-peer networks in which nodes have a copy of the blockchain and all connections are fast and reliable. Users can verify malicious behavior instantly without trusting third parties by simply synchronizing their blockchains and verifying the latest blocks. In our network architecture, however, light nodes can disconnect for long periods or even indefinitely. Device downtime is especially challenging for use cases where the direction of payments is biased towards a light node, such as when a seller uses her/his device to receive transactions from multiple buyers. In this case, we expect border payment channels, i.e., channels between light nodes and full nodes, to be highly imbalanced towards one of the parties. The initial channel capacity will be imbalanced

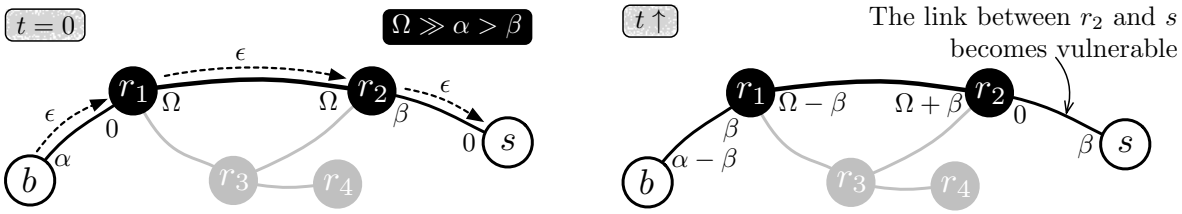

Figure 4. An example of the token theft vulnerability in WPCNs. On the left, a continuous amount of ϵ token flows from buyer b to seller s until the channel between r_2 and s is depleted. Then, on the right, s becomes highly vulnerable if it loses connection before closing the channel because r_2 has nothing to lose by closing the channel with a previous balance.

towards the light node if the node is a buyer and towards the full node if the node represents a seller.

Imbalanced channels in resource-constrained environments enhance the token theft problem. Let two resource-constrained devices, b and s, represent devices from a buyer and a seller, respectively, and be connected to entry nodes r_1 and r_2 via unidirectional payment channels as shown in Figure 4. Each payment channel $u \leftrightarrow v$ has a balance $bal_{u \leftrightarrow v}(t) = (bal_u(t), bal_v(t))$, where $bal_u(t)$ and $bal_v(t)$ are the balances of nodes u and v at time t, respectively. Note that $bal_u(t) + bal_v(t)$ is constant. For edge payment channels between buyers and entry nodes, e.g. $b \leftrightarrow r_1$, we assume an initial balance of $bal_{b \leftrightarrow r_1}(0) = (\alpha, 0)$, where α is an amount of tokens that buyer b reserves for payments in the channel. Likewise, the initial balance of edge payment channels between sellers and entry nodes, e.g. $r_2 \leftrightarrow s$ is $bal_{r_2 \leftrightarrow s}(0) = (\beta, 0)$ where β is the amount of tokens the entry node r_2 reserves for routing payments to the seller s. We assume for simplicity and w.l.o.g. that s and b only participate in a single payment channel.

Once a payment $\mathsf{pay}(\langle b, s\rangle, \{r_1, r_2\}, \epsilon)$ of ϵ tokens occurs from b to s in this configuration, r_2 and s sign a commitment transaction $Tx(1)$ containing the new balance of channel $bal_{r_2 \leftrightarrow s}(1) = (\beta - \epsilon, \epsilon)$. If s disconnects indefinitely at this moment, r_2 can close the channel with the operation $\mathtt{closeChannel}(\langle r_2, s\rangle, Tx(0))$ and recover ϵ tokens. Doing so is risky because r_2 would lose all its tokens if s recovers and detects the malicious behavior before the lock time window. However, as s receives more payments, the balance in $r_2 \leftrightarrow s$ will converge to $bal_{r_2 \leftrightarrow s}(t) = (0, \beta)$. Once this happens, r_2 has nothing to lose by closing the channel with a previous transaction even if s recovers on time. This is the optimal strategy for any rational entry node R once a border payment channel to a seller becomes depleted. Malicious nodes may also decide to attack intermediary cases depending on the risk-benefit ratio. Hence, we must modify the traditional security mechanisms of PCNs to prevent full nodes from adopting this strategy. Otherwise, the seller s is prone to token theft even in the absence of malicious behavior.

Note that even though we formulate the problem for an extreme case where a node is either a buyer or a seller, it still applies to any situation in which a light node receives payments and disconnects without closing the channel. The problem does not apply to buyers because the other party can only lose tokens by publishing a previous transaction. For a generic situation in which light nodes act as buyers and sellers, e.g., in a trading fair, every light node becomes vulnerable as soon as it receives a payment.

4 Defining a Minimum Time Window

A simple solution to the token theft problem is to hire "watchtower" nodes that are always online and constantly verify the blockchain to detect channels that have been improperly closed [2, 5, 6]. The solution, however, implies the light node must disclose all commitment transactions to the watchtower and trust it not to act maliciously. A second countermeasure would be to create a reputation system for full nodes in which light nodes would punish malicious behavior by issuing feedbacks. However, reputation systems lead to centralization and introduce new attack vectors that would be difficult to handle in decentralized environments [15].

Instead of adopting watchtowers or creating a reputation system, we propose a simple statistical approach: discover a lock time window W that minimizes the chance of attacks. Since most payment channel networks already adopt lock time windows as a security measure [5–7], we believe this solution is the easiest to implement and possibly the most impactful. To the best of our knowledge, no works have ever proposed to find a minimum time lock window value to prevent attacks. Our contribution also serves traditional PCNs as a guideline for users to select the best window parameters based on their connectivity patterns.

Our proposal. The lock time window W, defined in the number of blocks on each channel creation operation, represents the amount of time that tokens must remain locked until one party can claim them. The lock time window serves as a countermeasure against attacks based on unilateral closing of channels. For example, suppose a node on a channel disconnects and a token theft attack occurs. In that case, the

attacked party has W blocks to recover, verify the blockchain, and punish the attacker before she/he claims the tokens. Hence, the larger W is, the more secure the channel becomes against token theft. Conversely, setting a large W can create bottlenecks in routing and punish honest nodes that wish to close the channel correctly after the other party disconnects. In such cases, W should be as small as possible to not lock channel capacities for long periods and improve overall throughput. Therefore, the value of W represents a trade-off between security and efficiency, and our goal is to minimize W while guaranteeing a minimum level of security.

Let s be a resource-constrained device. We propose a methodology that uses four parameters to estimate W:

(i) T_{off}, a random variable that models the time s remains disconnected from the system, which can occur due to device failure or packet loss. T_{off} can either be modeled through a continuous random distribution or be estimated with empirical data from a dataset.

(ii) D_{det}, a random variable that models the delay for s to detect the attack. D_{det} follows a Poisson distribution with expected value bound by the equation

$$E[D_{\text{det}}] = n\frac{E[T_{\text{off}}]}{b_t}\left(\frac{b_s}{d} + v\right), \tag{1}$$

where n is the number of full nodes from which s requests blocks, $E[T_{\text{off}}]$ is the average downtime of s, b_t is the block time[2], b_s is the average block size, d is the average download rate of s, v is the average time it takes for s to verify all transactions in a block. In this equation, $\frac{E[T_{\text{off}}]}{b_t}$ represents the number of lost blocks.

(iii) D_{pun}, a random variable that models the delay for s to punish malicious behavior after detection. As punishment incurs publishing a transaction in the blockchain, the distribution of D_{pun} follows the Poisson distribution defined by Nakamoto [16] with expected value

$$E[D_{\text{pun}}] = n_c b_t, \tag{2}$$

where n_c is the number of confirmations it takes for a transaction to be valid and b_t is the block time. Note that D_{pun} is directly proportional to the assumptions of the underlying blockchain. The default values would be approximately $n_c = 6$ and $b_t = 600s$ for Bitcoin [16], $n_c = 20$ and $b_t = 15s$ for Ethereum, etc.

(iv) Δ, a random variable that estimates the relative bias of each open payment channel in the network. For empirical data, we calculate the imbalance ratio of each channel in the dataset using the equation

$$\Delta_i = \left|\frac{bal_u - bal_v}{bal_u + bal_v}\right| \forall \langle u, v\rangle \in \mathbb{E}(t), \tag{3}$$

where i is the channel index and bal_u and bal_v are the balances of the channel counterparties. The balance Δ_i serves as an estimation of how vulnerable to token theft the channel is, as we expect a trend to occur towards the party who has less capacity.

Finally, each of the four described parameters compose the equation of W:

$$W = (T_{\text{off}} + D_{\text{det}} + D_{\text{pun}})(1 + \Delta). \tag{4}$$

The rationale behind our definition is that the lock time window W must be at least $T_{\text{off}} + D_{\text{det}} + D_{\text{pun}}$, otherwise the victim cannot recover and punish the attacker before she/he spends the stolen tokens. Besides, we increase the minimum lock time window by a factor of Δ to account for the extra vulnerability of imbalanced channels. We assume the more imbalanced a channel initially is, the more payments should occur from the node with more capacity to the node with less capacity. Hence, by using Δ as part of the equation, we can increase the channel's security proportionally to the trend of flows we expect to occur. If the channel is balanced, we expect both parties to transact with each other evenly, and thus neither will be interested in attacking the channel. Note that because T_{off}, D_{det}, D_{pun}, and Δ are either model distributions or real statistical data, the value of W will also be a random distribution. The actual value of W to be selected by a user depends on what level of security she/he wishes to adopt for her/his use case. Users who invest heavily in the channel should select higher W thresholds because being attacked incurs great losses. Users who are willing to risk losses can select smaller thresholds to provide token liquidity.

5 Prototype Analysis and Results

We use real data from the Lightning Network [5] as a reference since it is the most adopted PCN [2]. We also use mobile broadband (MBB) connections as our profile connectivity model for wireless devices due to its massive presence [17]. However, the methodology we propose is agnostic to blockchains, connectivity profiles, and payment channel topologies. It suffices to estimate the four parameters described in Section 4 to find a minimum safe value for the time window that addresses any specific use case. We provide the complete code of our implementation on GitHub[3].

5.1 Evaluation Setup

We simulate three different scenarios that correspond to real availability measurements of mobile broadband devices to estimate the distribution of T_{off} [18, 19]. For the high-availability case, we use the downtime and packet loss distributions of MBB connections as measured by Elmokashfi *et al.* [18]. In their work, more than 90% of connections use 4G technology and the average downtime of an connection is 86.4s per day. We use the work from Baltrunas *et al.* [19] as reference for the medium-availability case. The work measures the availability of MBB connections that use 3G as the

[2]The block time is the average time it takes to mine a block in the system.

[3]In case of acceptance, we will provide the code at: https://github.com/gfrebello/pcn-time-window

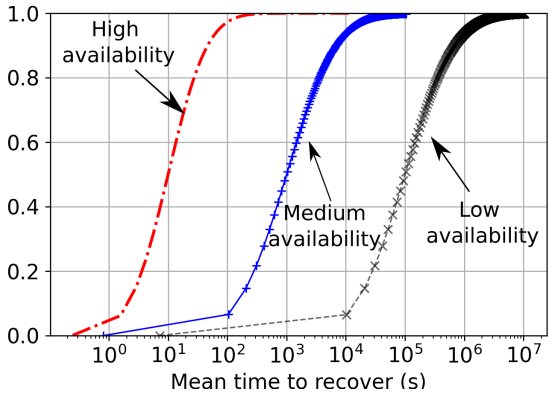

Figure 5. Cumulative density functions of T_{off}, the downtime of a vulnerable node. We model the distributions according to real data from previous studies on 3G and 4G mobile broadband networks [18, 19].

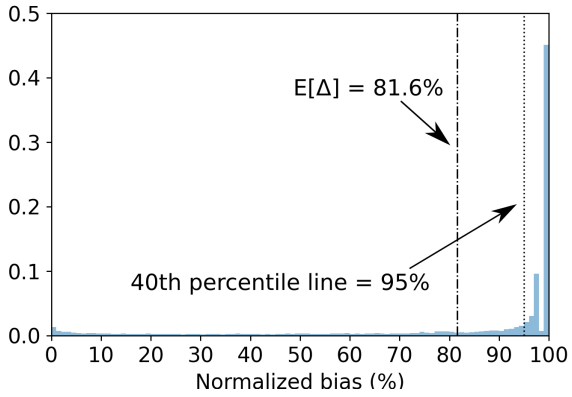

Figure 6. Normalized bias Δ of payment channels in the Lightning Network. 60% of channels present over 95% bias towards one party and the average bias is 81%, which indicates a heavily asymmetric behavior of payment flows.

default technology and shows that the downtime can last for hours every day. Lastly, we simulate a low-availability scenario by shifting the medium-availability distribution to the right by the average distance between the high-availability and medium-availability distributions. This yields an average downtime of about one week. Figure 5 depicts all cases side by side. By simulating three roughly symmetrical scenarios based on real data, we can predict how different levels of availability impact the minimum lock time window. This could be extended to real-world device data of any kind.

For the detection delay D_{det}, we set the parameters as $n = 3$, $b_t = 600s$, and $b_s = 10Mb/s$. $n = 3$ represents the minimum number of different nodes to request blocks to in case one node is faulty, and b_s and b_t follow the average block time and block size of Bitcoin, respectively. $E[T_{\text{off}}]$ is calculted according to the corresponding scenario and the download rate d is set using previous MBB evalutations: $d = 30Mb/s$ for high avaliability, $d = 2Mb/s$ for medium availability, and $d = 1Mb/s$ for low availability [19].

The parameters for the punishment delay D_{pun} come from Bitcoin. $b_t = 600s$ is the average block time and n_c, the number of confirmations, is set according to two different scenarios. In the optimistic scenario, we assume a transaction is confirmed as soon as it appears in a block; in the default scenario, we use the 6-confirmation rule proposed by Nakamoto in the Bitcoin paper [16].

5.2 Analysis and Discussion

We extract the values of Δ from LNChannels[4], an open-source Lightning Network explorer that offers a data set of the Lightning Network. We download the channel balances from all closed channels since the beginning of the network and calculate the normalized bias Δ_i of each channel according to Equation 3. Figure 6 depicts the Δ distribution. First, we observe an asymmetric trend of payment flows towards one party even though the distribution is composed of channel balances of a traditional PCN. This confirms the token theft problem is not exclusive to wireless PCNs. Hence, adopting minimum lock time windows that depend on average channel imbalances may fit a wide range of PCN models. The behavior is also coherent with previous works that analyze the Lightning Network [20, 21]. Second, there is a gap around the 99% percentile that is likely due to the Lightning Network implementation. BOLT#2 of the Lightning Network documentation [4] a user-defined minimum payment amount called `dust_limit_satoshis` that, if not met, invalidates the transaction and transforms it into channel fees. We believe the default value of `dust_limit_satoshis` creates the gap in the Δ distribution by not allowing parties to pay when the channel is almost depleted.

Finally, we evaluate W through thresholds $W(p)$ that correspond to the necessary W value for a device to detect and punish an attacker with p probability. Hence, a user that adopts $W(p)$ obtains p probability of being safe and assumes $(1 - p)$ probability of being attacked successfully. We use $W(50\%)$ as a reference for an insecure threshold and $W(95\%)$ for a secure threshold and measure the trade-off between insecure and secure time windows by calculating the distance d between the two thresholds. Short distances mean no significant gain from adopting a smaller window, while long distances mean the trade-off is significant. Therefore, the user should carefully select the value of W according to her/his needs. Figure 7 depicts the cumulative density functions for the minimum window sizes of all the considered scenarios. The thresholds $W(p)$ are equivalent to the percentiles of the distribution of W.

[4] Available at https://ln.fiatjaf.com/

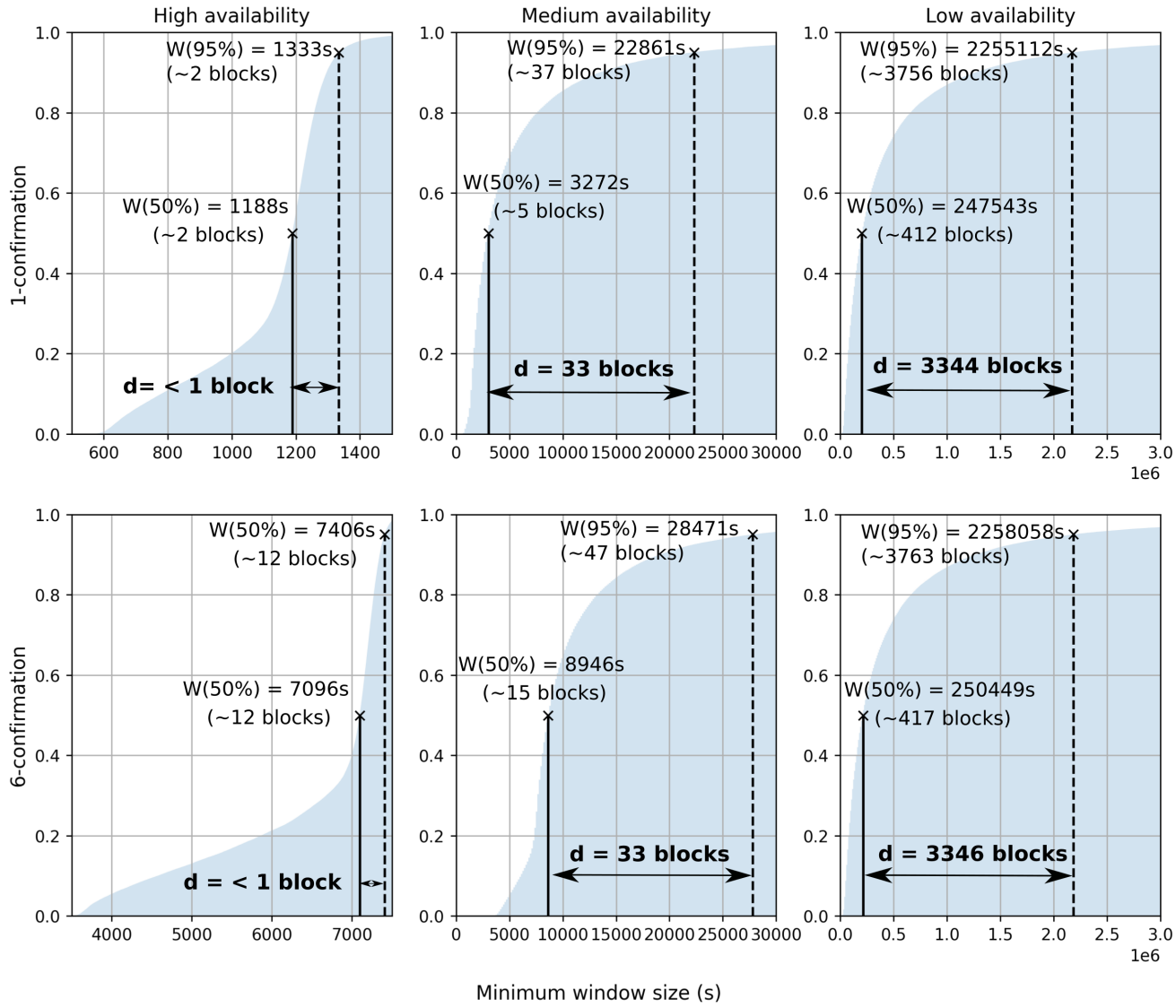

Figure 7. Lock time window sizes for all levels of availability with single confirmation and 6-block confirmation. When the availability is high, the distance d between the 50% and 95% thresholds remains below one block time, which indicates a small window is safe for most users. For medium and low availability, the distance increases significantly and forces the user to select a time window that better fits her/his security and delay needs.

In the high-availability scenario, 4G connectivity allows devices to be safe from attacks even with short time windows. The distance of less than one block between $W(50\%)$ and $W(95\%)$ demonstrates that increasing W to a secure level generates no significant delay, so devices with good connectivity should adopt the safest W possible. The result also confirms our assumption that good connectivity profiles present in most traditional PCNs can mitigate token theft.

The trade-off between security and efficiency becomes significant in the medium-availability scenario. The distance of 33 blocks between $W(50\%)$ and $W(95\%)$ corresponds to an increase of 5.5 hours in return delays for the party that closes the channel. As the time to recover increases, the differences between the 1-confirmation and 6-confirmation W values decrease. T_{off} becomes the dominant parameter in Equation 4. The results indicate that a user with 3G connectivity should define minimum lock time windows of at least a few hours to reduce the probability of attacks; otherwise, attackers with better connectivity can easily exploit them.

The low-availability scenario demonstrates that users with low connectivity should either select W values in the range of days to weeks or use the main blockchain to transact. Delays in such order of magnitude may be economically worthwhile if the fees to publish transactions in the blockchain are too

expensive for the user. However, assuming a 6-confirmation delay, more than 550 transactions could be published within the distance of 3346 blocks. The time window W may not be the most efficient countermeasure for devices that remain offline for long periods. Instead, we should adopt W with other security features, such as heavier punishment for attackers.

6 Related Work

Several works propose adaptations of traditional PCNs for mobile devices. Kurt *et al.* propose LNGate, a lightweight protocol for IoT devices to use Bitcoin's Lightning Network [5] via untrusted gateways [8]. Hannon *et al.* propose a similar protocol for the Lightning Network and demonstrate its security and fairness using game theory [9]. Robert *et al.* propose an integration of the Lightning Network with existing large-scale IoT ecosystems [10]. Mercan *et al.* present alternative lightweight PCN implementations that focus on reducing the computational needs for mobile devices [22]. The works, however, focus on adapting the Lightning Network to IoT scenarios. We propose a new PCN design and a security feature that is agnostic to PCN implementations.

Other works analyze the security of traditional PCNs. Mizrahi *et al.* [23] and Tochner *et al.* [24] formulate topology-based attacks that aim to disrupt the routing protocol of traditional PCNs. Erdin *et al.* compare the security and privacy of several PCN implementations and identify emerging attack vectors [2]. The works neither discuss attacks in PCNs with resource-constrained devices nor present efficient countermeasures for wireless environments. To the best of our knowledge, our work is the first to propose an architecture for wireless PCNs, formulate the token theft attack and propose a time window analysis as an efficient countermeasure.

7 Conclusion

We proposed a hybrid architecture that allows resource-constrained wireless nodes to issue off-chain transactions and analyzed the impact of the token theft problem in such environments. Our main findings show that the problem is not exclusive to wireless PCNs and that our solution may work with traditional PCNs as well. A countermeasure based on minimum lock time windows is efficient when the devices present high to medium availability. For devices with low availability, the minimum lock time window becomes so significant that it may be better to publish the transactions directly in the blockchain.

In future works, we will investigate other types of countermeasures to token theft, such as more efficient punishment mechanisms and time-varying lock time windows.

References

[1] G. A. F. Rebello, G. F. Camilo, L. C. Guimarães, L. A. C. de Souza, and O. C. M. Duarte, "On the security and performance of proof-based consensus protocols," in *4th CIoT*. IEEE, 2020, pp. 67–74.

[2] E. Erdin, S. Mercan, and K. Akkaya, "An evaluation of cryptocurrency payment channel networks and their privacy implications," *arXiv preprint arXiv:2102.02659*, 2021.

[3] Blockchain.com, "Blockchain charts," 2021, Last access: 25 June 2021. [Online]. Available: https://www.blockchain.com/charts

[4] J. Poon and O. Osuntokun, "BOLT #2: Peer protocol for channel management," 2021, Last access: 25 June 2021. [Online]. Available: https://github.com/lightningnetwork/lightning-rfc/blob/master/02-peer-protocol.md

[5] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," 2016, Last access: 25 June 2021. [Online]. Available: https://www.bitcoinlightning.com/wp-content/uploads/2018/03/lightning-network-paper.pdf

[6] brainbot labs Est., "The raiden network: Fast, cheap, scalable token transfers for ethereum," 2020, Last access: 25 June 2021. [Online]. Available: https://raiden.network/

[7] S. Roos, P. Moreno-Sanchez, A. Kate, and I. Goldberg, "Settling payments fast and private: Efficient decentralized routing for path-based transactions," *arXiv preprint arXiv:1709.05748*, 2017.

[8] A. Kurt, S. Mercan, O. Shlomovits, E. Erdin, and K. Akkaya, "Lngate: Powering iot with next generation lightning micro-payments using threshold cryptography," *arXiv preprint arXiv:2105.08902*, 2021.

[9] C. Hannon and D. Jin, "Bitcoin payment-channels for resource limited iot devices," in *IEEE COINS*, 2019, pp. 50–57.

[10] J. Robert, S. Kubler, and S. Ghatpande, "Enhanced lightning network (off-chain)-based micropayment in iot ecosystems," *Future Generation Computer Systems*, vol. 112, pp. 283–296, 2020.

[11] L. Lys, A. Micoulet, and M. Potop-Butucaru, "Atomic cross chain swaps via relays and adapters," in *3rd CryBlock*, 2020, pp. 59–64.

[12] V. Sivaraman, S. B. Venkatakrishnan, K. Ruan, P. Negi, L. Yang, R. Mittal, G. Fanti, and M. Alizadeh, "High throughput cryptocurrency routing in payment channel networks," in *17th USENIX NSDI*, 2020, pp. 777–796.

[13] E. Rohrer and F. Tschorsch, "Counting down thunder: Timing attacks on privacy in payment channel networks," in *ACM AFT*, 2020, pp. 214–227.

[14] G. Malavolta, P. Moreno-Sanchez, A. Kate, M. Maffei, and S. Ravi, "Concurrency and privacy with payment-channel networks," in *2017 ACM SIGSAC CCS*, 2017, pp. 455–471.

[15] G. F. Camilo, G. A. F. Rebello, L. A. C. de Souza, and O. C. M. Duarte, "A secure personal-data trading system based on blockchain, trust, and reputation," in *3rd IEEE Blockchain*. IEEE, 2020, pp. 379–384.

[16] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008, Last access: 25 June 2021. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[17] OECD, "Mobile broadband subscriptions indicator," 2021, Last access: 25 June 2021. [Online]. Available: https://doi.org/10.1787/1277ddc6-en

[18] A. Elmokashfi, D. Zhou, and D. Baltrünas, "Adding the next nine: An investigation of mobile broadband networks availability," in *23rd ACM MobiCom*, 2017, pp. 88–100.

[19] D. Baltrunas, A. Elmokashfi, and A. Kvalbein, "Measuring the reliability of mobile broadband networks," in *14th ACM IMC*, 2014, pp. 45–58.

[20] S. Tikhomirov, R. Pickhardt, A. Biryukov, and M. Nowostawski, "Probing channel balances in the lightning network," *arXiv preprint arXiv:2004.00333*, 2020.

[21] Y. Guo, J. Tong, and C. Feng, "A measurement study of bitcoin lightning network," in *2nd IEEE Blockchain*. IEEE, 2019, pp. 202–211.

[22] S. Mercan, E. Erdin, and K. Akkaya, "Improving sustainability of cryptocurrency payment networks for iot applications," in *IEEE ICC Workshops*. IEEE, 2020, pp. 1–6.

[23] A. Mizrahi and A. Zohar, "Congestion attacks in payment channel networks," *arXiv preprint arXiv:2002.06564*, 2020.

[24] S. Tochner, A. Zohar, and S. Schmid, "Route hijacking and dos in off-chain networks," in *2nd ACM AFT*. ACM, 2020, p. 228–240.