FPPW: A Fair and Privacy Preserving Watchtower For Bitcoin

Arash Mirzaei, Amin Sakzad, Jiangshan Yu, and Ron Steinfeld

Faculty of Information Technology, Monash University, Australia
{arash.mirzaei,amin.sakzad,jiangshan.yu,ron.steinfeld}@monash.edu

Abstract. In this paper, we introduce FPPW, a new payment channel with watchtower scheme for Bitcoin. This new scheme provides fairness w.r.t. all channel participants including both channel parties and the watchtower. It means that the funds of any honest channel participant are safe even assuming that other two channel participants are corrupted and/or collude with each other. Furthermore, the watchtower in FPPW learns no information about the off-chain transactions and hence the channel balance privacy is preserved. As a byproduct, we also define the coverage of a watchtower scheme, that is the total capacity of channels that a watchtower can cover on a scale of 0 to 1, and show that FPPW's coverage is higher than those of PISA and Cerberus. The scheme can be implemented without any update in Bitcoin script.

Keywords: Bitcoin · Security · Privacy · Payment channel · Lightning network · Generalized channel · Watchtower

1 Introduction

Scalability has always been an important limitation of Bitcoin. Payment channel is a promising technique to resolve this issue. It enables two parties to open a channel by locking some funds in a 2-of-2 multi-signature output. Then parties can update the channel state by exchanging off-chain transactions. Finally, they record the last agreed state on-chain and each party receives its deserved amount of funds accordingly. Since off-chain transactions are not recorded on the blockchain, payment channels also provide some privacy guarantees.

Lightning [16] and generalized channels [1] are two important payment channels for Bitcoin. In a Lightning channel, commit transactions represent the channel states and each party has its own version of the transaction. However, in a generalized channel, both parties hold the same version of the state and adaptor signature is effectively used to distinguish the broadcaster of the transaction from its counter-party. Several schemes also exist on Turing complete blockchains (e.g. Ethereum) [6, 8, 15].

Most payment channels work based on this idea that once a dishonest channel party records an old state on-chain, its counter-party is supposed to provide evidence of invalidity of the published state within a time interval. Otherwise, the channel gets finalized with the recorded old state. Since duration of this time interval is limited, such payment channels rely on the assumption that channel parties check the blockchain frequently. However, since it is possible that channel parties crash or go offline for a long time, they might delegate the monitoring task to a third-party, called the watchtower. Monitor [5], DCWC [3], Outpust [10], Cerberus [4], and Tee Guard [12] are the existing watchtower schemes for Bitcoin, where Tee Guard follows a different direction as it relies on features of Trusted Execution Environments. There are also some watchtower schemes for Turing complete blockchains [2, 13, 14].

Monitor [5] is the first watchtower scheme for Lightning network which mainly focuses on the channel privacy against watchtower. However, Monitor has two main issues, both of which related to fairness. Firstly, honest watchtowers might be rewarded upon fraud (i.e. broadcast of an old state on-chain), which is unfair with respect to (w.r.t.) the watchtower. Secondly, honest parties cannot penalize the unresponsive watchtower, which is unfair towards an honest hiring party.

DCWC [3] proposes the usage of a network of watchtowers which must cooperate to maximize their interest. This reduces the probability that the channel gets finalized with an old state. However, watchtowers might still crash or get unresponsive without being penalized by the hiring party. Also, the reward mechanism is still unfair w.r.t. the watchtower. Outpost [10] solves the issue of fairness towards the watchtower by paying her per channel update.

Cerberus [4] and PISA [14] elegantly provide fairness w.r.t. the hiring party. However, PISA fails to be deployed in cryptocurrencies with limited script languages such as Bitcoin and Cerberus sacrifices the channel balance privacy. In particular, the Cerberus watchtower learns the distribution of funds in the channel. Thus, the main motivation of this paper is designing a watchtower scheme for Bitcoin that achieves both: (1) fairness w.r.t. both the hired watchtower and her hiring party and (2) channel balance privacy.

1.1 Our Contribution

The contribution of this paper is as follows:

- We present a new privacy-preserving payment channel with watchtower scheme for Bitcoin called FPPW, which is fair w.r.t. all channel participants and allows the channel parties to go offline for a long period of time (Section 4). To be more precise, FPPW is an extension of a new variant of generalized channel.
- We are the first to define the concepts of fairness, channel balance privacy and coverage for a watchtower service (Section 3.3), where coverage is a metric that represents the maximum total capacity of channels that the watchtower can cover on a scale of 0 to 1. Furthermore, in Section 5, we prove that our design achieves fairness w.r.t. all channel participants and unlike Cerberus, it provides channel balance privacy. We also show that the coverage of FPPW is better than that of Cerberus and PISA. Table 1 presents a quick comparison between FPPW and other schemes.

Table 1. Comparison of FPPW with existing watchtower schemes.

Scheme	Bitcoin Support	Balance Privacy	β-Coverage[a]	Channel Party α-Fairness[b]	Watchtower Fairness
Monitor [5]	Yes	Yes	$\beta = 1$	$\alpha = 0$	No
DCWC [3]	Yes	Yes	$\beta = 1$	$\alpha = 0$	No
Outpost [10]	Yes	Yes	$\beta = 1$	$\alpha = 0$	Yes
PISA [14]	No	Yes	$\beta = 1/3$	$\alpha = 1$[c]	Yes
Cerberus [4]	Yes	No	$\beta = 1/3$	$\alpha = 1$	Yes
FPPW (this work)	Yes	Yes	$\beta = 1/2$	$\alpha = 1$	Yes

[a]: β is a value between 0 and 1 where the higher β, the higher achievable total capacity for the channels that are monitored by the watchtower.

[b]: α is a value between 0 and 1 and represents that the channel party might lose $(1 - \alpha)$ portion of its balance if the watchtower is unresponsive.

[c]: PISA allows the watchtower to lock an agreed amount of collateral per customer (i.e. $0 < \alpha \leq 1$). PISA also provides β-coverage with $\beta = 1/(1 + 2\alpha)$. However, in this table, to compare coverage of PISA with those of Cerberus and FPPW, we let PISA's collateral to be equal to the channel capacity (i.e. $\alpha = 1$).

– We propose a fee handling mechanism that allows the channel participants to determine the fee for different transactions at the time when fraud occurs. Furthermore, a proof-of-concept implementation of FPPW channels on Bitcoin is provided. [1]

2 Preliminaries and Notations

2.1 Preliminaries

In this section the underlying cryptographic primitives of FPPW are introduced.

Digital Signature A digital signature scheme Π includes three algorithms as following:

– **Key Generation.** $(pk, sk) \leftarrow \mathsf{Gen}(1^\kappa)$ on input 1^κ (κ is the security parameter), outputs the public/private key pair (pk, sk).
– **Signing.** $\sigma \leftarrow \mathsf{Sign}_{sk}(m)$ on inputs the private key sk and a message $m \in \{0, 1\}^*$ outputs the signature σ.
– **Verification.** $b \leftarrow \mathsf{Vrfy}_{pk}(m; \sigma)$ takes the public key pk, a message m and a signature σ as input and outputs a bit b.

In this work, we assume that the utilized signature schemes are existentially unforgeable under an adaptive chosen-message attack. It guarantees that the probability that an adversary who has access to a signing oracle outputs a valid signature on any new message is negligible. In this paper, we call such signature

[1] Due to lack of space, this will be presented in Appendix E.

schemes secure. ECDSA [9] is a secure signature scheme that is currently being used in Bitcoin. Schnorr [17] is another important secure signature scheme that has been proposed to be introduced in Bitcoin due to its key aggregation and signature aggregation properties.

Hard relation A relation $\mathcal{R}$ with statement/witness pairs $(Y;y)$ is called a hard relation if (i) There exists a polynomial time generating algorithm $(Y;y) \leftarrow \mathsf{GenR}(1^\kappa)$ that on input 1^κ outputs a statement/witness pair $(Y;y) \in \mathcal{R}$; (ii) The relation between Y and y can be verified in polynomial time, and (iii) For any polynomial-time adversary $\mathscr{A}$, the probability that $\mathscr{A}$ on input Y outputs y is negligible. We also let $L_{\mathcal{R}} := \{Y \mid \exists Y \ s.t. \ (Y,y) \in \mathcal{R}\}$. Statement/witness pairs of $\mathcal{R}$ can be public/private key of a signature scheme generated by Gen algorithm.

Adaptor Signature Adaptor signatures appeared first in [1]. Adaptor signature is used in generalized channels to tie together the authorization of a commit transaction and the leakage of a secret value. In what follows, we recall how an adaptor signature works. Given a hard relation $\mathcal{R}$ and a signature scheme $\mathit{\Pi}$, an adaptor signature protocol $\mathit{\Xi}$ includes four algorithms as follows:

- **Pre-Signing.** $\tilde{\sigma} \leftarrow \mathsf{pSign}_{sk}(m, Y)$ is a probabilistic polynomial time (PPT) algorithm that on input a private key sk, message $m \in \{0,1\}^*$ and statement $Y \in L_{\mathcal{R}}$, outputs a pre-signature $\tilde{\sigma}$.
- **Pre-Verification.** $b \leftarrow \mathsf{pVrfy}_{pk}(m, Y; \tilde{\sigma})$ is a deterministic polynomial time (DPT) algorithm that on input a public key pk, message $m \in \{0,1\}^*$, statement $Y \in L_{\mathcal{R}}$ and pre-signature $\tilde{\sigma}$, outputs a bit b.
- **Adaptation.** $\sigma \leftarrow \mathsf{Adapt}(\tilde{\sigma}, y)$ is a DPT algorithm that on input a pre-signature $\tilde{\sigma}$ and witness y, outputs a signature σ.
- **Extraction,** $\mathsf{Ext}(\sigma, \tilde{\sigma}, Y)$ is a DPT algorithm that on input a signature σ, pre-signature $\tilde{\sigma}$, and statement $Y \in L_{\mathcal{R}}$, outputs $\perp$ or a witness y such that $(Y, y) \in \mathcal{R}$.

Correctness of an adaptor signature guarantees that for an honestly generated pre-signature $\tilde{\sigma}$ on the message m w.r.t. a statement $Y \in L_{\mathcal{R}}$, we have $\mathsf{pVrfy}_{pk}(m, Y; \tilde{\sigma}) = 1$. Furthermore, when $\tilde{\sigma}$ is adapted to the signature σ, we have $\mathsf{Vrfy}_{pk}(m; \sigma) = 1$ and $\mathsf{Ext}(\sigma, \tilde{\sigma}, Y)$ outputs y such that $(Y, y) \in \mathcal{R}$.

An adaptor signature scheme is secure if it is existentially unforgeable under chosen message attack ($\mathsf{aEUF{-}CMA}$ security), pre-signature adaptable and witness extractable. The $\mathsf{aEUF{-}CMA}$ security guarantees that it is of negligible probability that any PPT adversary who has access to signing and pre-signing oracles outputs a valid signature for any arbitrary new message m even given a valid pre-signature and its corresponding Y on m. Pre-signature adaptablity guarantees that every pre-signature (possibly generated maliciously) w.r.t. Y can adapt to a valid signature using the witness y with $(Y, y) \in \mathcal{R}$. Witness extractablity guarantees that it is of negligible probability that any PPT adversary who has access to signing and pre-signing oracles outputs a valid signature and a statement Y for any new message m such that the valid signature does not reveal a witness for Y even given a valid pre-signature on m w.r.t. Y. The ECDSA-based and Schnorr-based adaptor signature schemes were constructed and analyzed in [1].

2.2 Notations

In this section, we present the notations for Bitcoin transactions. A Bitcoin transaction Tx_i has some inputs and some outputs and is denoted by:

$$Tx_i = [I_i^1, I_i^2, \ldots] \rightarrow [O_i^1, O_i^2, \ldots],$$

where I_i^j and O_i^j with $j \geq 0$ denote the j^{th} input and the j^{th} output of Tx_i, respectively. If Tx_i has one output, this output is denoted by O_i. Each Bitcoin output O, denoted by $(x \mid \varphi)$, consists of a monetary value x and some conditions φ that must be met when one takes O as an input in another transaction. If an output has several subconditions, they are separated by $\vee$ operation(s); To spend the output, one of the subconditions must be met. Each transaction input I has also two elements where the first one is actually the output of a previously published transaction O and the second element is the witness γ that I uses to meet the condition φ of O. The witness γ has also two elements, first of which is denoted by S and determines the index of the subcondition that I meets. The second element of γ, which is denoted by D, is actually the data that is required to meet the subcondition. To simplify the notations, we denote I by $(O \| S)$. If a transaction Tx_i lacks some required data in witness part of at least one of its inputs, it is denoted by $[Tx_i]$.

The signature and pre-signature of party $\mathcal{P}$ on Tx_i for its j^{th} input is denoted by $\sigma_i^{\mathcal{P},j}$ and $\tilde{\sigma}_i^{\mathcal{P},j}$, respectively, where j can be removed if Tx_i has one input. Since transaction flows might be difficult to follow, we also use charts to illustrate them. For example, $Tx_i = [(O_j^3 \| 2), (O_k \| 1)] \rightarrow [O_i^1, O_i^2, \ldots]$ is illustrated in Fig. 1. Transactions that are already published on-chain are illustrated by doubled edge rectangles (e.g. Tx_j in Fig. 1). Transactions that are ready to be published are illustrated by single edge rectangles (e.g. Tx_i). Dotted edge rectangles show transactions that still lack the required witness for at least one input and hence are unprepared to be propagated in the blockchain network (e.g. $[Tx_k]$ in Fig. 1).

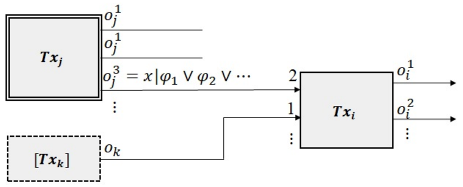


Fig. 1. Transaction flow of $Tx_i = [(O_j^3 \| 2), (O_k \| 1)] \rightarrow [O_i^1, O_i^2, \ldots]$.

For some transactions, the output that is taken as input to the transaction is irrelevant to protocol design. Such inputs are notated by $(x \mid \#_{\mathcal{P}})$ where x and $\mathcal{P}$ denotes the value and owner of that taken output, respectively. For example, the funding transaction of a payment channel between $\mathcal{A}$ and $\mathcal{B}$ is denoted by:

$$Tx_i = [(a \mid \#_{\mathcal{A}}), (b \mid \#_{\mathcal{B}})] \rightarrow [(a + b \mid pk_{\mathcal{A}} \wedge pk_{\mathcal{B}})].$$

Table 2 summarizes the mentioned notations.

Table 2. Notations

Notation	Description
Tx_i	Transaction $Tx_i = [I_i^1, I_i^2, \ldots] \rightarrow [O_i^1, O_i^2, \ldots]$ with inputs I_i^1, I_i^2, etc and outputs O_i^1, O_i^2, etc.
I_i^j	j^{th} input of transaction Tx_i
O_i^j	j^{th} output of Tx_i. Index j can be removed if Tx_i has a single output.
$O = (x \mid \varphi)$	Output with monetary value x and condition φ
γ_i^j	Witness of the input I_i^j
$\sigma_i^{\mathcal{P},j}$ (or $\tilde{\sigma}_i^{\mathcal{P},j}$)	Signature (or pre-signature) of $\mathcal{P}$ on j^{th} input of Tx_i. The index j can be removed if Tx_i has a single input.
$I = (O\|\|S)$	The input that meets S^{th} subcondition of the output O
$[Tx_i]$	Transaction Tx_i with incomplete witness for at least one input
$(x\|\#_A)$	Any arbitrary output owned by $\mathcal{A}$ with monetary value of x

3 FPPW Overview

3.1 System Model

Cryptographic primitives that have been used in FPPW are cryptographically secure. There is an authenticated and secure end-to-end communication channel between channel parties. The watchtower and channel parties are rational and might deviate from the protocol if it increases their profit. Also, each pair of participants might collude with each other if it raises the total profit of colluding participants. The watchtower is an always online service provider, but channel parties can go offline for a long period (approximately T rounds). Furthermore, the underlying blockchain contains a distributed ledger that achieves security [7]. When a valid transaction is propagated in the blockchain network, it is definitely included in the blockchain ledger immediately (i.e. the confirmation delay τ is 1).

Remark 1. FPPW channels can work with any confirmation delay. However, we assume that the confirmation delay is 1 to simplify the protocol and its analysis.

3.2 FPPW Overview

A payment channel contains a sequence of state updates between two parties where only its first and last states are recorded on the blockchain. The two channel parties process all the intermediate state updates off-chain. This eliminates the need to confirm every state update, i.e. every transaction, on the blockchain. However, as one may submit an intermediate state (which is already revoked by a later state) to the blockchain, the channel parties will need to get online frequently to monitor and punish such misbehaviours. Such a requirement may be impractical for some users. Thus, watchtower is introduced as a third party to act on behalf of the channel parties.

FPPW is a fair and privacy preserving watchtower service for generalised channels [1]. FPPW provides channel balance privacy and hence the watchtower obtains no data on intermediate state updates. To provide fairness towards the watchtower, the FPPW service rewards the watchtower for the channel establishment and per channel update. Furthermore, to achieve fairness w.r.t. channel parties, the watchtower must lock some collateral, which is redeemed if the watchtower is responsive upon fraudulent channel closures. If the watchtower is dishonest and the channel is closed at an old state, protocol guarantees that the cheated party can penalize the watchtower by taking its collateral. Watchtower can reclaim its collateral at any time. Then, the channel parties can update the channel on-chain and hire a new watchtower or continue using the channel. In the latter case, channel parties must get online frequently.

3.3 Watchtower Service Properties

In this section, some properties of a watchtower service are formally defined.

Definition 1 (Channel party α-Fairness). *A payment channel with watchtower is α-party-fair, if the following holds for an honest channel party $\mathcal{P}$:*

- *$\mathcal{P}$ can close the channel at any time and*
- *α is the largest real number such that regardless of the reward that $\mathcal{P}$ pays to the watchtower, $\mathcal{P}$ loses at most $(1-\alpha)\cdot x_{\mathcal{P}}$ coins in the channel where $x_{\mathcal{P}}$ denotes balance of $\mathcal{P}$ in the latest channel state.*

Note that $0 \leq \alpha \leq 1$, where $\alpha = 1$ implies that the honest party $\mathcal{P}$ will not lose any fund in the channel and $\alpha = 0$ means that $\mathcal{P}$ might lose all of his funds.

Definition 2 (Watchtower Fairness). *A payment channel with watchtower is watchtower-fair, if the following holds for an honest watchtower $\mathcal{W}$:*

- *$\mathcal{W}$ is rewarded with some non-zero amounts of coins and*
- *given that $\mathcal{W}$ has locked some collateral as part of the watching service, it is of negligible probability that the honest watchtower cannot redeem all the collateral once watching terminates according to the watching agreement.*

Monitor [5] and DCWC [3] are called unfair w.r.t the watchtower because for these schemes, it is possible that the watchtower is not rewarded.

Let $x_{\mathcal{P},0}$ with $\mathcal{P} \in \{\mathcal{A}, \mathcal{B}\}$ denote the initial balance of party $\mathcal{P}$ in the channel and the channel capacity be defined as $X := x_{\mathcal{A},0} + x_{\mathcal{B},0}$. The *Weak channel balance privacy* is defined by the following *privacy game*.

Challenge. Let there exist two payment channels where the first one is between honest channel parties $\mathcal{A}$ and $\mathcal{B}$ and the second one is between honest channel parties $\mathcal{A}'$ and $\mathcal{B}'$ and both channels have the same channel capacity X and the same number of channel updates n. Let $\mathbf{x}_{\mathcal{P},[i,j]}$ show the sequence of balance values of party $\mathcal{P}$ between i^{th} to j^{th} states of the payment channel that $\mathcal{P}$ is involved in. Assume that $\mathscr{A}$ is any PPT adversary excluding $\mathcal{A}$, $\mathcal{B}$, $\mathcal{A}'$ and $\mathcal{B}'$. To challenge the adversary, the challenger selects a random bit b and gives the sequence $(\mathbf{x}_{\mathcal{P},[1,n-1]}, \mathbf{x}_{\bar{\mathcal{P}},[1,n-1]})$ to $\mathscr{A}$ where $\mathcal{P} = \mathcal{A}$ and $\bar{\mathcal{P}} = \mathcal{B}$ if $b = 0$ and $\mathcal{P} = \mathcal{A}'$ and $\bar{\mathcal{P}} = \mathcal{B}'$ otherwise.

Output. The adversary $\mathscr{A}$ outputs a bit b' to guess that the received sequence belongs to the first or the second channel. The adversary wins the game if and only if $b = b'$.

Definition 3 (Weak Channel Balance Privacy). *A payment channel provides weak channel balance privacy if according to the privacy game* $\mid \Pr[b = b'] - 1/2 \mid$ *is negligible.*[2]

Next, we define β-coverage, which basically measures the capability of a watchtower (on a scale between 0 to 1) in watching all the existing payment channels on a fixed Blockchain.

Definition 4 (Coverage). *For a blockchain $\mathbb{B}$ with N payment channels, a watchtower $\mathcal{W}$ provides β-coverage with $\beta := \frac{\mathcal{X}}{\mathcal{C}+\mathcal{X}}$, where $\mathcal{C}$ is the total collateral required by $\mathcal{W}$ to watch all payment channels for both channel parties and $\mathcal{X}$ is the total capacity of all channels.*

The parameter β can take any value in the interval $[0, 1]$. For Cerberus and PISA (with $\alpha = 1$), β equals $\frac{1}{3}$ because for these schemes, collateral of the watchtower must be twice the channel capacity if the watchtower is going to be hired by both channel parties. Although, PISA allows lower values of collateral, such values cannot provide channel party α-fairness with $\alpha = 1$ and hence cannot guarantee that the honest party does not lose any funds.

4 FPPW Channel

The lifetime of an FPPW channel can be divided into 4 phases including establishment, update, closure and abort. We explain these phases through the

[2] In the challenge phase, if the sequence $(\mathbf{x}_{\mathcal{P},[0,n]}, \mathbf{x}_{\bar{\mathcal{P}},[0,n]})$ is given to the adversary, the defined privacy is called strong channel balance privacy. Monitor, DCWC, and Outpost provides strong channel balance privacy, while PISA provides weak channel balance privacy and Cerberus does not achieve channel balance privacy.

following sections. The cryptographic primitives, used in these phases, are as following: A digital signature scheme $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$; a hard relation $\mathcal{R}$ with generating algorithm $\mathsf{GenR} = \mathsf{Gen}$; an adaptor signature scheme $\Xi_{\Pi,\mathcal{R}} = (\mathsf{pSign}, \mathsf{pVrfy}, \mathsf{Adapt}, \mathsf{Ext})$. We assume that the watchtower is hired by both channel parties. However, FPPW can be simply extended to situations where only one party hires the watchtower. FPPW for such scenarios will be provided in Appendix F.

4.1 FPPW Channel Establishment

FPPW channel establishment phase includes a funding transaction, a commit transaction and a split transaction. The funding transaction locks funds of the channel parties in a 2-of-2 multisig output and can be claimed only if both parties agree and cooperate with each other. The commit transaction is held by both channel parties and sends all the channel funds to a joint account that can be spent by the corresponding split transaction after t rounds. Split transaction actually represents the channel state and distributes the channel funds between the channel parties. The quantity t, which is called the revocation period, exists to ensure that there is enough time for punishing the dishonest channel party in case of fraud (i.e. if the published commit transaction corresponds with a revoked state). Parties finally publish the funding transaction on the blockchain. However, since its output can be spent if both parties cooperate, one party might lock the funds by being unresponsive. To avoid such situations, before signing the funding transaction, both channel parties must sign commit and split transactions.

Additionally, two other transactions are created in this phase including the collateral transaction and the reclaim transaction which are used for watchtower services. Using the collateral transaction, the watchtower locks its collateral in a 3-of-3 multisig output shared between channel parties and the watchtower. Collateral is awarded to the cheated channel party if the watchtower does not appropriately act upon fraud. The value of the collateral equals the channel capacity. Using the reclaim transaction, the watchtower can start the process of reclaiming its collateral. The watchtower can finally redeem its collateral by claiming the output of the reclaim transaction after a large relative timelock of T rounds with $T \gg t$ which is called the penalty period. If channel parties get online at least once every $T-1$ rounds, they will always have enough time to take the dishonest watchtower's collateral as compensation and prevent an unresponsive watchtower from redeeming its collateral. However, if the honest watchtower has published the reclaim transaction to withdraw its service, channel parties will have two options. They can either update the channel on-chain with a new watchtower or remain almost always online to prevent from fraudulent channel closures. Collateral transaction is finally recorded on-chain. However, to avoid any hostage situation, before publishing the collateral transaction, the watchtower must receive channel parties' signatures on the reclaim transaction.

All the above-mentioned transactions are further explained hereinafter.

- **Funding transaction**: Using this transaction, channel parties $\mathcal{A}$ and $\mathcal{B}$ open an FPPW channel. Funding transaction is defined as follows:

$$Tx_{\mathbb{FU}} := [(a + \epsilon/2 \mid \#_{\mathcal{A}}), (b + \epsilon/2 \mid \#_{\mathcal{B}})] \rightarrow [(a + b + \epsilon \mid (pk_{\mathcal{A}} \wedge pk_{\mathcal{B}}))], \quad (1)$$

  where ϵ is the minimum value supported by the Bitcoin blockchain and a and b are the initial balance of $\mathcal{A}$ and $\mathcal{B}$ in the channel (regardless of the negligible value $\epsilon/2$). Output of $Tx_{\mathbb{FU}}$ is a 2-of-2 multisig output shared between $\mathcal{A}$ and $\mathcal{B}$. The public keys $pk_{\mathcal{A}}$ and $pk_{\mathcal{B}}$ of $\mathcal{A}$ and $\mathcal{B}$ are generated using the key generation algorithm of the underlying digital signature Gen.
- **Commit transaction**: There exists one commit transaction $Tx_{\mathbb{CM},i}$ per state but only the first one ($Tx_{\mathbb{CM},i}$ with $i = 0$) is created at the channel establishment phase. $Tx_{\mathbb{CM},i}$ is as follows:

$$Tx_{\mathbb{CM},i} := [(O_{\mathbb{FU}} \| 1)] \rightarrow [(a + b \mid \varphi^1_{\mathbb{CM},i}), (\epsilon \mid \varphi^2_{\mathbb{CM},i})], \quad (2)$$

  where $\varphi^1_{\mathbb{CM},i} := \varphi^1_{\mathbb{CM},i(1)} \vee \varphi^1_{\mathbb{CM},i(2)}$ with $\varphi^1_{\mathbb{CM},i(1)} := pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge \Delta_t$, $\varphi^1_{\mathbb{CM},i(2)} := pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge pk_{\mathcal{W}}$ and $\varphi^2_{\mathbb{CM},i} := \varphi^2_{\mathbb{CM},i(1)} \vee \varphi^2_{\mathbb{CM},i(2)} \vee \varphi^2_{\mathbb{CM},i(3)}$ with $\varphi^2_{\mathbb{CM},i(1)} := pk_{\mathcal{B}} \wedge Y_{\mathcal{A},i} \wedge \Delta_t$, $\varphi^2_{\mathbb{CM},i(2)} := pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge pk_{\mathcal{W}}$ and $\varphi^2_{\mathbb{CM},i(3)} := pk_{\mathcal{A}} \wedge Y_{\mathcal{B},i} \wedge \Delta_t$ where $Y_{\mathcal{A},i}$ and $Y_{\mathcal{B},i}$ are statements of a hard relation $\mathcal{R}$ generated by $\mathcal{A}$ and $\mathcal{B}$ for the ith state using the generating algorithm GenR and Δ_t shows relative timelock of t rounds. $O^1_{\mathbb{CM},i}$ is the main output with value of $a + b$. Normally, if parties act honestly and $Tx_{\mathbb{CM},i}$ is published on-chain, the first subcondition of its main output ($pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge \Delta_t$) is met by $Tx_{\mathbb{SP},i}$ after t rounds. The $O^2_{\mathbb{CM},i}$ with value of ϵ is the auxiliary output, which as will be explained in Section 4.2, is only used for watchtower purposes.
  The transaction $Tx_{\mathbb{CM},i}$ requires signatures of both parties $\mathcal{A}$ and $\mathcal{B}$ to be published. To generate $\sigma^{\mathcal{B}}_{\mathbb{CM},i}$, party $\mathcal{A}$ generates a statement/witness pair $(Y_{\mathcal{A},i}, y_{\mathcal{A},i})$ and sends the statement $Y_{\mathcal{A},i}$ to $\mathcal{B}$. Then, party $\mathcal{B}$ uses the pre-signing algorithm pSign of the adaptor signature and $\mathcal{A}$'s statement $Y_{\mathcal{A},i}$ to generate a pre-signature $\tilde{\sigma}^{\mathcal{B}}_{\mathbb{CM},i}$ on $[Tx_{\mathbb{CM},i}]$ and sends the result to $\mathcal{A}$. Thus, whenever it is necessary, $\mathcal{A}$ is able to use the adaptation algorithm adapt of the adaptor signature to transform the pre-signature to the signature $\sigma^{\mathcal{B}}_{\mathbb{CM},i}$ and publish $Tx_{\mathbb{CM},i}$ on-chain. This also enables $\mathcal{B}$ to use the extraction algorithm $\mathsf{Extract}$, the published signature and its corresponding pre-signature to extract the witness value $y_{\mathcal{A},i}$. The witness value, as will be seen in Section 4.2, might be used to punish a dishonest channel party by claiming all the channel funds or to penalize an unresponsive watchtower.

Remark 2. $\mathcal{A}$ has two public keys in $O^1_{\mathbb{CM},i}$, which for simplicity, we denote them both by $pk_{\mathcal{A}}$. However, in practice such public keys are selected disjointly. This is also extended to other participants and other outputs.

- **Split transaction**: $Tx_{\mathbb{SP},i}$ actually represents the ith channel state where only the first one ($Tx_{\mathbb{SP},i}$ with $i = 0$) is created in the channel establishment phase. This transaction is as follows:

$$Tx_{\mathbb{SP},i} := [(O^1_{\mathbb{CM},i} \| 1)] \rightarrow [(x^1_{\mathbb{SP},i} \mid \varphi^1_{\mathbb{SP},i}), (x^2_{\mathbb{SP},i} \mid \varphi^2_{\mathbb{SP},i}), \ldots]. \quad (3)$$

The $Tx_{\mathbb{SP},i}$ spends the main output of $Tx_{\mathbb{CM},i}$ by meeting the subcondition $pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge \Delta_t$.

- **Collateral transaction**: $Tx_{\mathbb{CL}}$ locks the collateral of the watchtower on-chain and its output can be spent if $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{W}$ cooperate. Value of collateral c equals $a+b$. The $Tx_{\mathbb{CL}}$ is defined as follows:

$$Tx_{\mathbb{CL}} := [(c \mid \#_{\mathcal{W}})] \rightarrow [(c \mid pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge pk_{\mathcal{W}})]. \tag{4}$$

- **Reclaim transaction**: This transaction spends the output of $Tx_{\mathbb{CL}}$ and its output can be spent by $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{W}$ if they cooperate or by $\mathcal{W}$ after a long relative timelock period. The $Tx_{\mathbb{RC}}$ is defined as follows:

$$Tx_{\mathbb{RC}} := [(O_{\mathbb{CL}} \| 1)] \rightarrow [(c \mid (pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge pk_{\mathcal{W}}) \vee (pk_{\mathcal{W}} \wedge \Delta_T))]. \tag{5}$$

  The second subcondition is used by the watchtower to redeem its collateral after T rounds and withdraw its service. However, as will be mentioned in following sections, the first subcondition is used to penalize the unresponsive watchtower.

Fig. 2 summarizes the channel establishment phase. Appendix C provides details of the corresponding protocol.

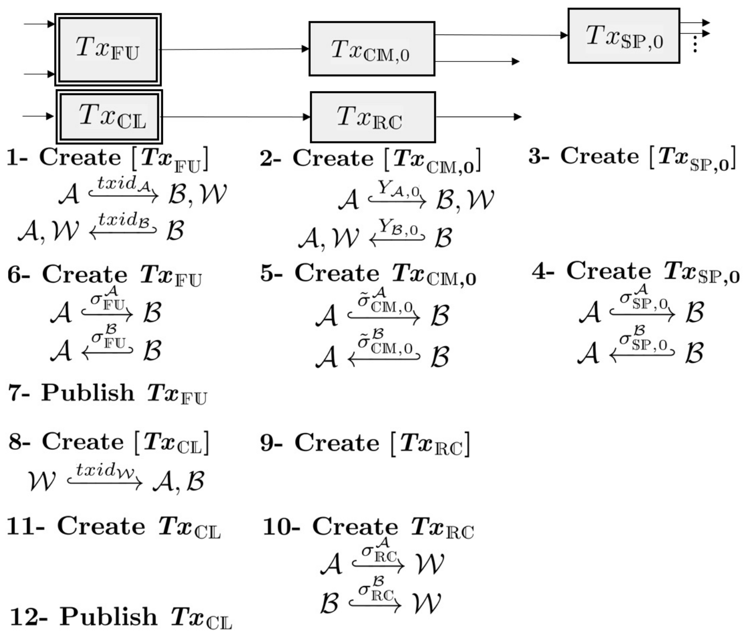

$Tx_{\mathbb{FU}}$
$Tx_{\mathbb{CL}}$
$Tx_{\mathbb{CM},0}$
$Tx_{\mathbb{RC}}$
$Tx_{\mathbb{SP},0}$
1- Create [$Tx_{\mathbb{FU}}$]
$\mathcal{A} \xrightarrow{txid_{\mathcal{A}}} \mathcal{B}, \mathcal{W}$
$\mathcal{A}, \mathcal{W} \xleftarrow{txid_{\mathcal{B}}} \mathcal{B}$
2- Create [$Tx_{\mathbb{CM},0}$]
$\mathcal{A} \xrightarrow{Y_{\mathcal{A},0}} \mathcal{B}, \mathcal{W}$
$\mathcal{A}, \mathcal{W} \xleftarrow{Y_{\mathcal{B},0}} \mathcal{B}$
3- Create [$Tx_{\mathbb{SP},0}$]
4- Create $Tx_{\mathbb{SP},0}$
$\mathcal{A} \xrightarrow{\sigma^{\mathcal{A}}_{\mathbb{SP},0}} \mathcal{B}$
$\mathcal{A} \xleftarrow{\sigma^{\mathcal{B}}_{\mathbb{SP},0}} \mathcal{B}$
5- Create $Tx_{\mathbb{CM},0}$
$\mathcal{A} \xrightarrow{\tilde{\sigma}^{\mathcal{A}}_{\mathbb{CM},0}} \mathcal{B}$
$\mathcal{A} \xleftarrow{\tilde{\sigma}^{\mathcal{B}}_{\mathbb{CM},0}} \mathcal{B}$
6- Create $Tx_{\mathbb{FU}}$
$\mathcal{A} \xrightarrow{\sigma^{\mathcal{A}}_{\mathbb{FU}}} \mathcal{B}$
$\mathcal{A} \xleftarrow{\sigma^{\mathcal{B}}_{\mathbb{FU}}} \mathcal{B}$
7- Publish $Tx_{\mathbb{FU}}$
8- Create [$Tx_{\mathbb{CL}}$]
$\mathcal{W} \xrightarrow{txid_{\mathcal{W}}} \mathcal{A}, \mathcal{B}$
9- Create [$Tx_{\mathbb{RC}}$]
10- Create $Tx_{\mathbb{RC}}$
$\mathcal{A} \xrightarrow{\sigma^{\mathcal{A}}_{\mathbb{RC}}} \mathcal{W}$
$\mathcal{B} \xrightarrow{\sigma^{\mathcal{B}}_{\mathbb{RC}}} \mathcal{W}$
11- Create $Tx_{\mathbb{CL}}$
12- Publish $Tx_{\mathbb{CL}}$

Fig. 2. A summary of FPPW Channel Establishment.

4.2 FPPW Channel Update

Assume that an FPPW channel is in state i with $i \geq 0$ and channel parties decide to update it from state i to $i+1$. This is performed in two sub-phases. In the first sub-phase, channel parties create a new commit transaction and a new split transaction for the new state. However, to avoid any hostage situation, they sign the split transaction before signing the commit transaction. In the second sub-phase, channel parties revoke the previous state by signing one revocation and two penalty transactions. At most one out of these three transactions might be published on-chain upon fraud (i.e. upon broadcast of the revoked commit transaction). While the revocation transaction might be used to penalize the cheating channel party, penalty transactions might be utilized for punishing the dishonest watchtower.

The revocation transaction is the only transaction that spends both outputs of the revoked commit transaction using their non-timelocked subconditions $pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge pk_{\mathcal{W}}$. Thus, once a dishonest channel party publishes the revoked commit transaction, the watchtower or the counter-party can immediately publish the revocation transaction. It invalidates both penalty transactions because they also spend the auxiliary output of the revoked commit transaction. The single output of the revocation transaction is spendable by someone who knows witness value y of both channel parties (i.e. party $\mathcal{A}$ can claim it if party $\mathcal{B}$ has published the revoked commit transaction and vice versa).

Now assume that a dishonest channel party publishes the revoked commit transaction but the watchtower does not react in time. Then the dishonest channel party might also publish the corresponding split transaction after t rounds. This spends the main output of the revoked commit transaction and invalidates the revocation transaction. However, since the honest channel party go offline for at most $T-1$ rounds, it gets online when the watchtower has not completed reclaiming its collateral yet (i.e. the watchtower has not broadcast the reclaim transaction or has not spent its output yet). Thus, the honest party can publish one of two penalty transactions. Both penalty transactions spend the auxiliary output of the revoked commit transaction as well as output of collateral and reclaim transaction, respectively. Similar to the revocation transaction, only the honest cheated party can claim output of the published penalty transaction.

The introduced transactions will be explained further bellow:

– **Revocation transaction**: When parties $\mathcal{A}$ and $\mathcal{B}$ want to revoke $Tx_{\mathbb{CM},i}$, each channel participant ($\mathcal{A}$, $\mathcal{B}$ and $\mathcal{W}$) generates all the required signatures for the revocation transaction $Tx_{\mathbb{RV},i}$ and sends the signatures to other two participants. $Tx_{\mathbb{RV},i}$ is as follows:

$$Tx_{\mathbb{RV},i} := [(O^1_{\mathbb{CM},i} \| 2), (O^2_{\mathbb{CM},i} \| 2)] \rightarrow [(a + b + \epsilon \mid Y_{\mathcal{A},i} \wedge Y_{\mathcal{B},i})]. \tag{6}$$

The $Tx_{\mathbb{RV},i}$ spends both outputs of $Tx_{\mathbb{CM},i}$ using the non-timelocked subcondition $pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge pk_{\mathcal{W}}$ and sends all the channel funds to an output with condition $Y_{\mathcal{A},i} \wedge Y_{\mathcal{B},i}$. When a dishonest party, let's say $\mathcal{A}$, publishes the revoked $Tx_{\mathbb{CM},i}$, $\mathcal{A}$ must wait for t rounds before being able to publish $Tx_{\mathbb{SP},i}$. However, $\mathcal{W}$ or $\mathcal{B}$ can immediately publish $Tx_{\mathbb{RV},i}$. Since $Tx_{\mathbb{CM},i}$ has been published by $\mathcal{A}$, party $\mathcal{B}$ can obtain $y_{\mathcal{A},i}$. Thus, only party $\mathcal{B}$ who knows both $y_{\mathcal{A},i}$ and $y_{\mathcal{B},i}$ will own all the channel funds.

– **Penalty transaction 1**: There is one penalty transaction 1 $Tx_{\mathbb{PN}_1,i}$ per revoked state which is used to penalize $\mathcal{W}$, given that a dishonest party publishes $Tx_{\mathbb{CM},i}$ and spends its main output using $Tx_{\mathbb{SP},i}$. The $Tx_{\mathbb{PN}_1,i}$ is defined as follows:

$$Tx_{\mathbb{PN}_1,i} := [(O^2_{\mathbb{CM},i}\|j), (O_{\mathbb{CL}}\|1)] \rightarrow [(c + \epsilon \mid Y_{\mathcal{A},i} \wedge Y_{\mathcal{B},i})], \tag{7}$$

where $j := 1$ given that broadcaster of $Tx_{\mathbb{CM},i}$ is $\mathcal{A}$ or $j := 3$ otherwise. When parties want to revoke $Tx_{\mathbb{CM},i}$, $\mathcal{A}$ and $\mathcal{W}$ ($\mathcal{B}$ and $\mathcal{W}$) compute the required signatures for the second input of $Tx_{\mathbb{PN}_1,i}$ and send the signatures to $\mathcal{B}$ ($\mathcal{A}$). Now assume that one party, let's say $\mathcal{A}$, publishes the revoked $Tx_{\mathbb{CM},i}$ and spends its main output after t rounds. Then, $\mathcal{B}$ obtains $y_{\mathcal{A},i}$ and hence can add the required signatures for the first input of $Tx_{\mathbb{PN}_1,i}$ and publish it, given that $O_{\mathbb{CL}}$ is still unspent. $Tx_{\mathbb{PN}_1,i}$ spends the second output of $Tx_{\mathbb{CM},i}$ using the timelocked subcondition $pk_{\mathcal{B}} \wedge Y_{\mathcal{A},i} \wedge \Delta_t$ as well as the output of the collateral transaction. Only $\mathcal{B}$ can claim output of $Tx_{\mathbb{PN}_1,i}$. A similar scenario can occur if $\mathcal{B}$ is the broadcaster of $Tx_{\mathbb{CM},i}$.

– **Penalty transaction 2**: There exists one penalty transaction 2 $Tx_{\mathbb{PN}_2,i}$ per state. It is exactly the same as $Tx_{\mathbb{PN}_1,i}$, with the only difference that it spends $O_{\mathbb{RC}}$ (rather that $O_{\mathbb{CL}}$) using the subcondition $pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge pk_{\mathcal{W}}$. Thus, it is useful for cases where the watchtower does not react upon fraud but by publishing $Tx_{\mathbb{RC}}$ tries to reclaim its collateral. However, since the honest party goes offline for at most $T-1$ rounds, it gets online when $O_{\mathbb{RC}}$ is still unspent. Thus, the honest party can add the required signatures to $[Tx_{\mathbb{PN}_2,i}]$ and publish it. The $Tx_{\mathbb{PN}_2,i}$ is defined as follows:

$$Tx_{\mathbb{PN}_2,i} := [(O^2_{\mathbb{CM},i}\|j), (O_{\mathbb{RC}}\|1)] \rightarrow [(c + \epsilon \mid Y_{\mathcal{A},i} \wedge Y_{\mathcal{B},i})], \tag{8}$$

where $j := 1$ given that broadcaster of $Tx_{\mathbb{CM},i}$ is party $\mathcal{A}$ or $j := 3$ otherwise.

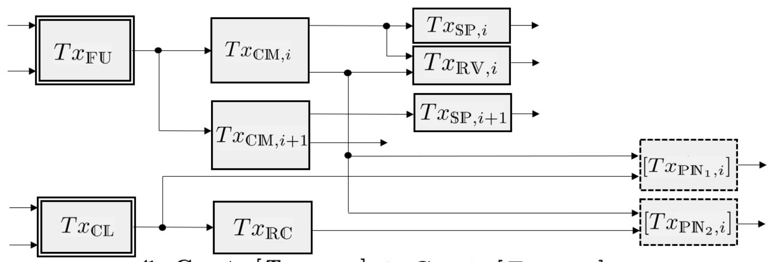
Fig. 3 summarizes the channel update phase. Appendix C provides details of the corresponding protocol.

Remark 3. Watchtower is actively involved in steps 6 and 7 of the channel update phase (See Fig. 3). Therefore, this phase fails to complete if the watchtower is unavailable. Appendix D introduces an update protocol for such scenarios.

4.3 FPPW Channel Closure

Assume that the channel parties $\mathcal{A}$ and $\mathcal{B}$ have updated their channel n times and then $\mathcal{A}$ and/or $\mathcal{B}$ decide to close it. They can close the channel cooperatively. To do so, $\mathcal{A}$ and $\mathcal{B}$ create a new transaction, called modified split transaction $Tx_{\overline{\mathbb{SP}}}$, and publish it on-chain. The $Tx_{\overline{\mathbb{SP}}}$ is defined as follows:

$$Tx_{\overline{\mathbb{SP}}} := [(O_{\mathbb{FU}}\|1)] \rightarrow [(x^1_{\overline{\mathbb{SP}}} \mid \varphi^1_{\overline{\mathbb{SP}}}), (x^2_{\overline{\mathbb{SP}}} \mid \varphi^2_{\overline{\mathbb{SP}}}), \ldots]. \tag{9}$$



Sub-phase 1

1- Create $[\boldsymbol{Tx}_{\mathbb{CM},i+1}]$ **2- Create** $[\boldsymbol{Tx}_{\mathbb{SP},i+1}]$

$\mathcal{A} \xleftrightarrow{Y_{\mathcal{A},i+1}} \mathcal{B}, \mathcal{W}$

$\mathcal{A}, \mathcal{W} \xleftrightarrow{Y_{\mathcal{B},i+1}} \mathcal{B}$

4- Create $\boldsymbol{Tx}_{\mathbb{CM},i+1}$ **3- Create** $\boldsymbol{Tx}_{\mathbb{SP},i+1}$

$\mathcal{A} \xleftrightarrow{\tilde{\sigma}^{\mathcal{A}}_{\mathbb{CM},i+1}} \mathcal{B}$ $\mathcal{A} \xleftrightarrow{\sigma^{\mathcal{A}}_{\mathbb{SP},i+1}} \mathcal{B}$

$\mathcal{A} \xleftrightarrow{\tilde{\sigma}^{\mathcal{B}}_{\mathbb{CM},i+1}} \mathcal{B}$ $\mathcal{A} \xleftrightarrow{\sigma^{\mathcal{B}}_{\mathbb{SP},i+1}} \mathcal{B}$

Sub-phase 2

5- Create $[\boldsymbol{Tx}_{\mathbb{RV},i}]$

6- Create $\boldsymbol{Tx}_{\mathbb{RV},i}$ **7- Create** $[\boldsymbol{Tx}_{\mathbb{PN}_1,i}], [\boldsymbol{Tx}_{\mathbb{PN}_2,i}]$

$\mathcal{W} \xleftrightarrow{\sigma^{\mathcal{W},1}_{\mathbb{RV},i}, \sigma^{\mathcal{W},2}_{\mathbb{RV},i}} \mathcal{A}, \mathcal{B}$ $\mathcal{W} \xleftrightarrow{\sigma^{\mathcal{W},2}_{\mathbb{PN}_1,i}, \sigma^{\mathcal{W},2}_{\mathbb{PN}_2,i}} \mathcal{A}, \mathcal{B}$

$\mathcal{A} \xleftrightarrow{\sigma^{\mathcal{A},1}_{\mathbb{RV},i}, \sigma^{\mathcal{A},2}_{\mathbb{RV},i}} \mathcal{B}, \mathcal{W}$ $\mathcal{A} \xleftrightarrow{\sigma^{\mathcal{A},2}_{\mathbb{PN}_1,i}, \sigma^{\mathcal{A},2}_{\mathbb{PN}_2,i}} \mathcal{B}$

$\mathcal{B} \xleftrightarrow{\sigma^{\mathcal{B},1}_{\mathbb{RV},i}, \sigma^{\mathcal{B},2}_{\mathbb{RV},i}} \mathcal{A}, \mathcal{W}$ $\mathcal{B} \xleftrightarrow{\sigma^{\mathcal{B},2}_{\mathbb{PN}_1,i}, \sigma^{\mathcal{B},2}_{\mathbb{PN}_2,i}} \mathcal{A}$

Fig. 3. FPPW Channel Update.

Outputs of this transaction might be similar to those for $Tx_{\mathbb{SP},n}$. Note that the value of auxiliary output of $Tx_{\mathbb{CM},n}$ (ϵ) can also be given to $\mathcal{A}$ and $\mathcal{B}$ ($\epsilon/2$ each) through outputs of $Tx_{\overline{\mathbb{SP}}}$. If one of the channel parties gets unresponsive, its counter-party can still close the channel non-collaboratively by publishing $Tx_{\mathbb{CM},n}$ and then $Tx_{\mathbb{SP},n}$ on-chain. The collaborative and non-collaborative channel closure protocols can be found in Appendix C.

It is always possible that a channel party publishes a revoked commit transaction $Tx_{\mathbb{CM},i}$ on-chain. Then, the watchtower or the counter-party publishes the corresponding revocation transaction within $t-1$ rounds. Only the honest counter-party can claim output of the revocation transaction. If the watchtower is unresponsive and the honest party is offline, a malicious party can publish a revoked commit transaction $Tx_{\mathbb{CM},i}$ with $i < n$ and its corresponding split transaction $Tx_{\mathbb{SP},i}$ on-chain. Then the honest party, who gets online once every $T-1$ rounds, can penalize the unresponsive watchtower by publishing either $Tx_{\mathbb{PN}_1,i}$ or $Tx_{\mathbb{PN}_2,i}$. Protocols for these scenarios can be found in Appendix C. Fig. 4 depicts transaction flows of FPPW.

4.4 FPPW Watchtower Abort

In this phase, $\mathcal{W}$ decides to terminate its employment by $\mathcal{A}$ and $\mathcal{B}$. To do this, $\mathcal{W}$ publishes $Tx_{\mathbb{RC}}$ and spends its output after T rounds. Since $\mathcal{A}$ and $\mathcal{B}$ do not go offline for more than $T-1$ rounds, they get online during this T-round inter-

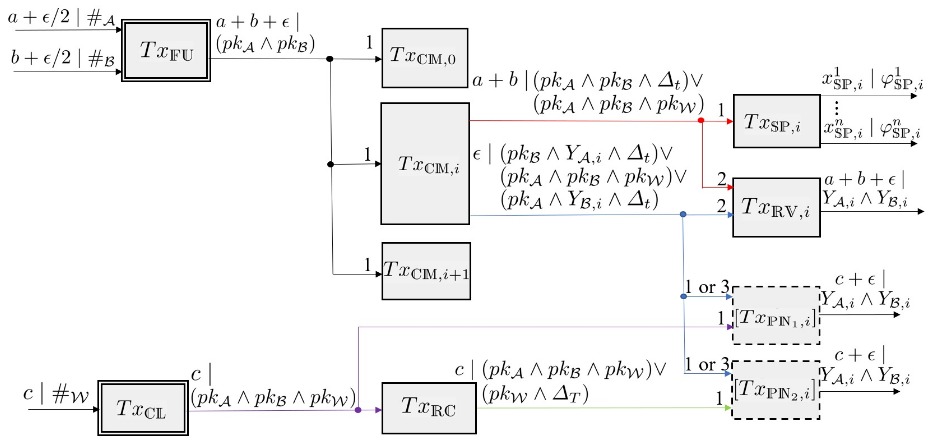

$a + \epsilon/2 \mid \#_{\mathcal{A}}$
$b + \epsilon/2 \mid \#_{\mathcal{B}}$
$Tx_{\mathbb{FU}}$
$a + b + \epsilon \mid (pk_{\mathcal{A}} \wedge pk_{\mathcal{B}})$
1
$Tx_{\mathbb{CM},0}$
$a + b \mid (pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge \Delta_t) \vee (pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge pk_{\mathcal{W}})$
1
$Tx_{\mathbb{CM},i}$
$\epsilon \mid (pk_{\mathcal{B}} \wedge Y_{\mathcal{A},i} \wedge \Delta_t) \vee (pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge pk_{\mathcal{W}}) \vee (pk_{\mathcal{A}} \wedge Y_{\mathcal{B},i} \wedge \Delta_t)$
1
$Tx_{\mathbb{CM},i+1}$
1
$Tx_{\mathbb{SP},i}$
$x^1_{\mathbb{SP},i} \mid \varphi^1_{\mathbb{SP},i}$
$x^n_{\mathbb{SP},i} \mid \varphi^n_{\mathbb{SP},i}$
2
2
$Tx_{\mathbb{RV},i}$
$a + b + \epsilon \mid Y_{\mathcal{A},i} \wedge Y_{\mathcal{B},i}$
1 or 3
1
$[Tx_{\mathbb{PN}_1,i}]$
$c + \epsilon \mid Y_{\mathcal{A},i} \wedge Y_{\mathcal{B},i}$
1 or 3
1
$[Tx_{\mathbb{PN}_2,i}]$
$c + \epsilon \mid Y_{\mathcal{A},i} \wedge Y_{\mathcal{B},i}$
$c \mid \#_{\mathcal{W}}$
$Tx_{\mathbb{CL}}$
$c \mid (pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge pk_{\mathcal{W}})$
$Tx_{\mathbb{RC}}$
$c \mid (pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge pk_{\mathcal{W}}) \vee (pk_{\mathcal{W}} \wedge \Delta_T)$

Fig. 4. An FPPW Bitcoin Channel.

val and observes that $Tx_{\mathbb{RC}}$ is on the chain. Then parties can close the channel and open a new one with a new watchtower. Parties can also continue using this channel without any watchtower. To do so, channel parties must check the blockchain at least once every $t - 1$ rounds to prevent from fraudulent channel closures. New channel updates can be performed according to generalized channels [1] or its new variant introduced in Appendix B.

5 Security Analysis

In this section, we analyze fairness, privacy and coverage of FPPW protocol through Theorems 1, 2, and 3, respectively. Lemmas 1 and 2 are used to prove Theorem 1. They show how FPPW guarantees that funds of the honest channel party and the honest watchtower are safe in the channel.

Lemma 1. *For an FPPW channel, assume that the honest channel party $\mathcal{P} \in \{\mathcal{A}, \mathcal{B}\}$ checks the blockchain at the end of the channel establishment phase and then gets online periodically with period of at most $T - 1$ rounds. The probability that $\mathcal{P}$ loses any funds in the channel is negligible.*

Although, proof of Lemma 1 will be presented in Appendix A, here we briefly discuss it. Without loss of generality let $\mathcal{P} = \mathcal{A}$. Cheating the honest party $\mathcal{A}$ using any scenario other than broadcast of a revoked commit transaction requires forging the signature of $\mathcal{A}$ and hence is of negligible probability. Channel establishment phase completes when $Tx_{\mathbb{CL}}$ is published on-chain. If $Tx_{\mathbb{CL}}$ is published through the block BL_j, the next time that $\mathcal{A}$ gets online, BL_{j+k} with $k \leq T - 1$ is the latest block on the chain and four possible events might have occurred regarding broadcast of a revoked $Tx_{\mathbb{CM},i}$ or $Tx_{\mathbb{RC}}$ during this interval:

- Case 1: When BL_{j+k} is the last block on the blockchain, $\mathcal{A}$ observes that only $Tx_{\mathbb{RC}}$ has been published on-chain. Consequently, $\mathcal{A}$ goes offline and checks the blockchain frequently with period of at most $t-1$ rounds. Now if a revoked $Tx_{\mathbb{CM},i}$ is published, it is of negligible probability that its outputs can be spent within $t-1$ rounds without $\mathcal{A}$'s authorization and $\mathcal{A}$ grants such an authorization only on $Tx_{\mathbb{RV},i}$. Also since $\mathcal{A}$ does not go offline for more than $t-1$ rounds, $\mathcal{A}$ will always have at least 1 round time to publish $Tx_{\mathbb{RV},i}$, which is enough according to our assumption regarding the value of the confirmation delay.
- Case 2: When BL_{j+k} is the last block on chain, $\mathcal{A}$ observes that both the revoked $Tx_{\mathbb{CM},i}$ and $Tx_{\mathbb{RC}}$ are on the chain. If fewer than $t-1$ blocks have published since broadcast of $Tx_{\mathbb{CM},i}$, $\mathcal{A}$ publishes $Tx_{\mathbb{RV},i}$. Otherwise, $\mathcal{A}$ will have at least 1 round time to publish $Tx_{\mathbb{PN}_2,i}$ which is enough according to our assumption regarding the value of the confirmation delay. The probability of other scenarios is negligible.
- Case 3: When BL_{j+k} is the last block on chain, $\mathcal{A}$ observes that $Tx_{\mathbb{CM},i}$ is on-chain but $Tx_{\mathbb{RC}}$ is unpublished. If fewer than $t-1$ blocks have published since broadcast of $Tx_{\mathbb{CM},i}$, party $\mathcal{A}$ can publish $Tx_{\mathbb{RV},i}$. Otherwise, $\mathcal{A}$ publishes $Tx_{\mathbb{PN}_1,i}$. If before publishing $Tx_{\mathbb{PN}_1,i}$, the transaction $Tx_{\mathbb{RC}}$ is recorded on-chain, $\mathcal{A}$ publishes $Tx_{\mathbb{PN}_2,i}$. Other scenarios happen with negligible probability.
- Case 4: When BL_{j+k} is the last block on chain, $\mathcal{A}$ observes that neither a revoked $Tx_{\mathbb{CM},i}$ nor $Tx_{\mathbb{RC}}$ are on-chain and goes offline for another $T-1$ rounds.

As it is obvious, Cases 1, 2, and 3 result in publishing either of $Tx_{\mathbb{RV},i}$, $Tx_{\mathbb{PN}_1,i}$ or $Tx_{\mathbb{PN}_2,i}$ on the chain. It is of negligible probability that broadcast of $Tx_{\mathbb{RV},i}$, $Tx_{\mathbb{PN}_1,i}$ or $Tx_{\mathbb{PN}_2,i}$ causes the honest party $\mathcal{A}$ to lose any funds in the channel because only $\mathcal{A}$ knows values of both $y_{\mathcal{A},i}$ and $y_{\mathcal{B},i}$. If Case 4 occurs, the process can repeat with all j being replaced with $j+k$.

Lemma 2. *For an FPPW channel, assume that the honest watchtower $\mathcal{W}$ checks the blockchain at the end of the channel establishment phase and then remains online. The probability that $\mathcal{W}$ loses any funds in the channel is negligible.*

As will be seen in the proof of Lemma 2 in Appendix A, an honest watchtower $\mathcal{W}$ does not lose any funds with non-negligible probability unless first a revoked commit transaction $Tx_{\mathbb{CM},i}$ is recorded on the blockchain and at least t rounds later, either $Tx_{\mathbb{PN}_1,i}$ or $Tx_{\mathbb{PN}_2,i}$ is also published on-chain. However, if a revoked $Tx_{\mathbb{CM},i}$ is published, it is of negligible probability that $O^1_{\mathbb{CM},i}$ or $O^2_{\mathbb{CM},i}$ are spent within $t-1$ rounds using any transaction other than $Tx_{\mathbb{RV},i}$. Thus, once $Tx_{\mathbb{CM},i}$ is published, $\mathcal{W}$ will have at least $t-1$ rounds time to publish $Tx_{\mathbb{RV},i}$ and invalidate both $Tx_{\mathbb{PN}_1,i}$ and $Tx_{\mathbb{PN}_2,i}$.

Theorem 1. *FPPW provides channel party α-fairness with $\alpha = 1$ and watchtower fairness as defined in Def. 1 and 2, respectively.*

Proof. The honest channel party always have at least one non-revoked commit transaction and its corresponding split transaction by broadcasting which she can close the channel. This proves that FPPW meets the first requirement of Def. 1. Furthermore, We know that based on FPPW protocol, the honest channel party checks the chain at least once every $T-1$ rounds and according to the above discussion (see Lemma 1), the probability that the honest channel party loses any funds in the channel is negligible. This proves that FPPW provides channel party α-fairness with $\alpha = 1$.

The watchtower in FPPW is paid for channel establishment and each channel update and hence her reward amount is non-zero. Also, we know that based on FPPW protocol, the honest watchtower always remains online and according to Lemma 2, the probability that such an honest watchtower loses any funds in the channel is negligible. Additionally, the watchtower can publish $Tx_{\mathbb{RC}}$ at any time and redeem her collateral after T rounds. Thus, FPPW meets both requirement of Def. 2.

Theorem 2. *FPPW provides weak channel balance privacy based on Def. 3.*

Proof. Assume that the conditions mentioned in the two-stage privacy game (see Section 3.3) are satisfied. By observing different steps and transactions of the protocol, one can see that only split transactions contain information on $x_{\mathcal{A},i}$ and $x_{\mathcal{B},i}$ with $i \in [1, n-1]$. However, these transactions are never published on-chain or sent to the watchtower or any external entity. Other transactions in the protocol contain no information regarding $x_{\mathcal{A},i}$ or $x_{\mathcal{B},i}$ with $i \in [1, n-1]$. Note that monetary value of $O^1_{\mathbb{CM},i}$, $O^2_{\mathbb{CM},i}$, $O^1_{\mathbb{RV},i}$, $O^1_{\mathbb{PN}_1,i}$, $O^1_{\mathbb{PN}_2,i}$, $O^1_{\mathbb{CL}}$, and $O^1_{\mathbb{RC}}$ of the first payment channels are the same as those for the second one. Furthermore, $Tx_{\mathbb{FU}}$, $Tx_{\mathbb{SP},n}$ or $Tx_{\overline{\mathbb{SP}}}$ contain no information regarding the i^{th} channel state with $i \in [1, n-1]$. Thus, the view of any adversary $\mathscr{A}$ on $(\mathbf{x}_{\mathcal{A},[1,n-1]}, \mathbf{x}_{\mathcal{B},[1,n-1]})$ is indistinguishable from its view on $(\mathbf{x}_{\mathcal{A}',[1,n-1]}, \mathbf{x}_{\mathcal{B}',[1,n-1]})$.

Theorem 3. *FPPW provides β-coverage with $\beta = 1/2$ based on Def. 4.*

Proof. Assume that we have N payment channels, with channel capacities $X_i = a_i + b_i$, $i \in [1, N]$. Thus, the total capacity of the channels is $\mathcal{X} = \sum_{i=1}^{N} X_i$. Since the i^{th} channel collateral c_i equals $a_i + b_i$, the total watchtower collateral is $\mathcal{C} = \sum_{i=1}^{N} c_i = \sum_{i=1}^{N} X_i = \mathcal{X}$. Thus, we have $\beta = \frac{\mathcal{X}}{\mathcal{X}+\mathcal{C}} = 1/2$.

6 Fee Handling

Once a revoked commit transaction is recorded on the blockchain, watchtower must record its corresponding revocation transaction within $t-1$ rounds. Otherwise the watchtower might be penalized. However, the time it takes for a transaction to be recorded on the blockchain depends on its fee value and the network congestion. Body of a revocation transaction is created during the channel update phase but it might be broadcast in the blockchain network later upon fraud. Thus, the fee amount must be large enough to ensure the watchtower that the revocation transaction will be accepted by miners within the revocation period. In other words, when channel participants are creating a revocation transaction, they must assume that the blockchain network will be highly congested at the time when fraud will occur.

An alternative approach is usage of SIGHASH of type 0x81 (SIGHASH_ALL | SIGHASH_ANYONECANPAY) for channel parties' signatures for both inputs of revocation transactions. Thus, signature for each input applies to that input and the output. Therefore, when due to network congestion the considered fee for the revocation transaction is low, the watchtower can add some inputs to the revocation transaction to increase the fee amount, sign all inputs using SIGHASH of type 0x01 (SIGHASH_ALL) and submit it to the network. If there exists enough time, the watchtower might even repeat this process several times and raise this extra fee each time until one of the revocation transactions is accepted by the miners. This method can be used if revocation transactions are only held by the watchtower (i.e. if channel parties do not receive signatures of the watchtower on revocation transactions during the channel update phase).

A similar approach can also be used for penalty transactions. Channel parties and the watchtower can use SIGHASH of type 0x02 (SIGHASH_NONE) for the second input of penalty transactions. Then, signatures apply only on all inputs of penalty transactions. In this way, the watchtower can be certain that a penalty transaction cannot be published unless its corresponding commit transaction is on-chain. However, if a revoked $Tx_{\mathbb{CM},i}$ is published by a channel party, let's say $\mathcal{A}$, and its main output is spent by $Tx_{\mathbb{SP},i}$, party $\mathcal{B}$ has the opportunity to set the output value of the penalty transaction according to the network congestion and sign the corresponding penalty transaction (to meet the subcondition $pk_{\mathcal{B}} \wedge Y_{\mathcal{A},i} \wedge \Delta_T$) using SIGHASH of type 0x01 (SIGHASH_ALL). In this way, $\mathcal{B}$ can reduce the output value if the current fee is low and this difference value is used as the extra fee amount. If there exists enough time, $\mathcal{B}$ can even repeat this process multiple times, each time with a higher fee until one penalty transaction is recorded on-chain.

References

1. Aumayr, L., Ersoy, O., Erwig, A., Faust, S., Hostakova, K., Maffei, M., Moreno-Sanchez, P., Riahi, S.: Generalized bitcoin-compatible channels. IACR Cryptol. ePrint Arch. **2020**, 476 (2020)
2. Avarikioti, G., Kogias, E.K., Wattenhofer, R.: Brick: Asynchronous state channels. arXiv preprint arXiv:1905.11360 (2019)
3. Avarikioti, G., Laufenberg, F., Sliwinski, J., Wang, Y., Wattenhofer, R.: Towards secure and efficient payment channels. arXiv preprint arXiv:1811.12740 (2018)
4. Avarikioti, G., Litos, O.S.T., Wattenhofer, R.: Cerberus channels: Incentivizing watchtowers for bitcoin. Financial Cryptography and Data Security (FC) (2020)
5. Dryja, T., Milano, S.B.: Unlinkable outsourced channel monitoring. Talk transcript) https://diyhpl. us/wiki/transcripts/scalingbitcoin/milan/unlinkable-outsourced-channel-monitoring (2016)

6. Dziembowski, S., Eckey, L., Faust, S., Malinowski, D.: Perun: Virtual payment hubs over cryptocurrencies. In: 2019 IEEE Symposium on Security and Privacy (SP). pp. 106–123. IEEE (2019)
7. Garay, J., Kiayias, A., Leonardos, N.: The bitcoin backbone protocol with chains of variable difficulty. In: Annual International Cryptology Conference. pp. 291–323. Springer (2017)
8. Green, M., Miers, I.: Bolt: Anonymous payment channels for decentralized currencies. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. pp. 473–489 (2017)
9. Johnson, D., Menezes, A., Vanstone, S.: The elliptic curve digital signature algorithm (ecdsa). International journal of information security **1**(1), 36–63 (2001)
10. Khabbazian, M., Nadahalli, T., Wattenhofer, R.: Outpost: A responsive lightweight watchtower. In: Proceedings of the 1st ACM Conference on Advances in Financial Technologies. pp. 31–40 (2019)
11. Lau, J., Wuille, P.: BIP: 143 (2016), https://github.com/bitcoin/bips/blob/master/bip-0143.mediawiki
12. Leinweber, M., Grundmann, M., Schönborn, L., Hartenstein, H.: Tee-based distributed watchtowers for fraud protection in the lightning network. In: Data Privacy Management, Cryptocurrencies and Blockchain Technology, pp. 177–194. Springer (2019)
13. Liu, B., Szalachowski, P., Sun, S.: Fail-safe watchtowers and short-lived assertions for payment channels. arXiv preprint arXiv:2003.06127 (2020)
14. McCorry, P., Bakshi, S., Bentov, I., Meiklejohn, S., Miller, A.: Pisa: Arbitration outsourcing for state channels. In: Proceedings of the 1st ACM Conference on Advances in Financial Technologies. pp. 16–30 (2019)
15. Miller, A., Bentov, I., Kumaresan, R., McCorry, P.: Sprites: Payment channels that go faster than lightning. CoRR abs/1702.05812 **306** (2017)
16. Poon, J., Dryja, T.: The bitcoin lightning network: Scalable off-chain instant payments (2016)
17. Schnorr, C.P.: Efficient signature generation by smart cards. Journal of cryptology **4**(3), 161–174 (1991)

A Omitted Proofs

As mentioned in Section 3.1, we assume that the underlying cryptographic primitives used in FPPW are cryptographically secure.

Lemma 3. *For an FPPW channel, it is of negligible probability that broadcast of $Tx_{\mathbb{RV},i}$, $Tx_{\mathbb{PN}_1,i}$ or $Tx_{\mathbb{PN}_2,i}$ causes the honest channel party $\mathcal{P} \in \{\mathcal{A}, \mathcal{B}\}$ to lose any funds in the channel.*

Proof. Without loss of generality let $\mathcal{P} = \mathcal{A}$. The transaction $Tx_{\mathbb{RV},i}$ spends the main output of the revoked $Tx_{\mathbb{CM},i}$ and hence cannot be published unless $Tx_{\mathbb{CM},i}$ is on-chain. Based on the protocol, the honest party $\mathcal{A}$ never broadcasts the revoked $Tx_{\mathbb{CM},i}$ on-chain. The party $\mathcal{A}$ only creates the pre-signature $\tilde{\sigma}_{\mathbb{CM},i}$ on the transaction $Tx_{\mathbb{CM},i}$. Thus, if $Tx_{\mathbb{CM},i}$ is published, the probability that $\mathcal{A}$ fails to obtain $y_{\mathcal{B},i}$ is negligible. Otherwise, $\mathsf{aEUF-CMA}$ security or witness extractability of the used adaptor signature is violated. Furthermore, $Tx_{\mathbb{RV},i}$

has only one output with the condition of $Y_{\mathcal{A},i} \wedge Y_{\mathcal{B},i}$ and the value of $a + b + \epsilon$. Since $\mathcal{A}$ privately preserves its witness value $y_{\mathcal{A},i}$, the probability that any PPT adversary claims $O_{\mathbb{RV},i}$ is negligible. Otherwise, the utilized hard relation would break. Therefore, it is of negligible probability that $\mathcal{A}$ (who knows both $y_{\mathcal{A},i}$ and $y_{\mathcal{B},i}$) fails to claim $O_{\mathbb{RV},i}$.

Also transactions $Tx_{\mathbb{PN}_1,i}$ and $Tx_{\mathbb{PN}_2,i}$ spend the auxiliary output of $Tx_{\mathbb{CM},i}$ as well as output of $Tx_{\mathbb{CL}}$ and $Tx_{\mathbb{RC}}$, respectively. These transactions have one output with the condition of $Y_{\mathcal{A},i} \wedge Y_{\mathcal{B},i}$ and the value of $c + \epsilon = a + b + \epsilon$. Since output condition for $Tx_{\mathbb{PN}_1,i}$ and $Tx_{\mathbb{PN}_2,i}$ is the same as that of $Tx_{\mathbb{RV},i}$, the proof for $Tx_{\mathbb{PN}_1,i}$ and $Tx_{\mathbb{PN}_2,i}$ is also similar to that of $Tx_{\mathbb{RV},i}$.

Remark 4. Before stating the next Lemma, it must be noted that while the channel update phase from state i to $i + 1$ is incomplete yet, it is possible that $Tx_{\mathbb{CM},i+1}$ is published on-chain and t rounds later $Tx_{\mathbb{SP},i+1}$ is also broadcast. In such situations we assume that the channel party does not lose any funds in the channel even if her counter-party's balance in state $i + 1$ is larger than that of state i. In other words, it is assumed that during steps 6 and 7 of the channel update phase (see Fig. 3), there are two valid channel states where channel closure with each one does not cause the honest party to lose any funds in the channel. A similar assumption is also made for other payment channels of type replace by revocation [16, 1].

Lemma 4. *For an FPPW channel with n channel updates, it is of negligible probability that the honest party $\mathcal{P} \in \{\mathcal{A}, \mathcal{B}\}$ loses any funds using any scenario other than broadcast of $Tx_{\mathbb{CM},i}$ with $i < n$.*

Proof. Without loss of generality let $\mathcal{P} = \mathcal{A}$. Funds of $\mathcal{A}$ are locked in $O_{\mathbb{FU}}$. It is of negligible probability that any PPT adversary $\mathscr{A}$ spends the output of $Tx_{\mathbb{FU}}$ without the honest party $\mathcal{A}$'s authorization. Otherwise, the underlying digital signature would be forgeable. Furthermore, $Tx_{\overline{\mathbb{SP}}}$, $Tx_{\mathbb{CM},i}$ with $i = [0, n - 1]$, $Tx_{\mathbb{CM},n}$ and possibly $Tx_{\mathbb{CM},n+1}$ (given that channel update phase from state n to $n + 1$ has started) are the only transactions in the protocol that spend the output of $Tx_{\mathbb{FU}}$ and $\mathcal{A}$ grants authorization for. Thus, these transactions will be discussed further to see how each one of them can cause the honest party $\mathcal{A}$ to be cheated out of its funds.

Since $Tx_{\overline{\mathbb{SP}}}$ represents the final agreed state of the channel, its broadcast cannot cause $\mathcal{A}$ to lose any funds in the channel. Additionally, since both sub-conditions of $O^1_{\mathbb{CM},n}$ include $pk_{\mathcal{A}}$, due to our assumption regarding the security of the underlying digital signature, it is of negligible probability that the main output of $Tx_{\mathbb{CM},n}$ is spent without $\mathcal{A}$'s authorization. Since $Tx_{\mathbb{SP},n}$ is the only transaction in the protocol that spends the main output of $Tx_{\mathbb{CM},n}$ and $\mathcal{A}$ grants authorization for, the probability of spending $O^1_{\mathbb{CM},n}$ using any transaction other than $Tx_{\mathbb{SP},n}$ is negligible. However, since $Tx_{\mathbb{SP},n}$ also represents the final state of the channel, its broadcast cannot cause $\mathcal{A}$ to lose any funds. According to Remark 4, similar statements can be stated for $Tx_{\mathbb{CM},n+1}$ and $Tx_{\mathbb{SP},n+1}$. Thus, cheating the honest party $\mathcal{A}$ using any scenario other than broadcast of $Tx_{\mathbb{CM},i}$ with $i < n$ is of negligible probability.

Case 1. Let there exist an FPPW channel with n channel updates where $n \geq 0$. Assume that the honest channel party $\mathcal{P} \in \{\mathcal{A}, \mathcal{B}\}$ gets online when the last published block on the blockchain is BL_j. Party $\mathcal{P}$ observes that $Tx_{\mathbb{CM},i}$ is unpublished but $Tx_{\mathbb{RC}}$ has been published through the block BL_k with $k \leq j$.

Lemma 5. *If conditions of Case 1 are satisfied, the probability that $\mathcal{P}$ loses any funds in the channel is negligible.*

Proof. Without loss of generality let $\mathcal{P} = \mathcal{A}$. If party $\mathcal{A}$ checks the blockchain and observes that $Tx_{\mathbb{CM},i}$ is unpublished but $Tx_{\mathbb{RC}}$ has been published, $\mathcal{A}$ can get online periodically with period of at most $t-1$ rounds. Based on Lemma 4, if $\mathcal{A}$ is going to lose some funds in the channel with non-negligible probability, the PPT adversary $\mathscr{A}$ must publish $Tx_{\mathbb{CM},i}$ with $i < n$ through the block BL_k with $k > j$. It must be noted that since the next time that $\mathcal{A}$ gets online again, BL_{j+t-1} is the latest block on the blockchain, if we have $k > j+t-1$, all the conditions mentioned for Case 1 repeat with j being replaced with $j+t-1$. Thus, we assume that we have $j < k \leq j+t-1$.

Timelocked subconditions of outputs of $Tx_{\mathbb{CM},i}$ cannot be met within $t-1$ rounds. Also, their non-timelocked subconditions include $pk_{\mathcal{A}}$. Therefore, it is of negligible probability that any adversary spends the first or the second output of $Tx_{\mathbb{CM},i}$ through one of the blocks $BL_{k+1}, \cdots, BL_{k+t-1}$ without the honest party $\mathcal{A}$'s authorization. Otherwise, the underlying digital signature would break. However, $\mathcal{A}$ never grants such authorizations on a transaction other than $Tx_{\mathbb{RV},i}$. Furthermore, the honest party $\mathcal{A}$ has created $Tx_{\mathbb{RV},i}$ through step 6 of the channel update phase from state i to $i+1$. When $\mathcal{A}$ gets online the last block on the blockchain is BL_{j+t-1}. Thus, $\mathcal{A}$ is able to publish $Tx_{\mathbb{RV},i}$ through one of the blocks $BL_{j+t}, \cdots, BL_{k+t-1}$. Since we have $k - j \geq 1$, $\mathcal{A}$ always have at least 1 block time to publish $Tx_{\mathbb{RV},i}$, which is enough based on our blockchain assumption regarding the value of the confirmation delay. Also, according to Lemma 3, it is of negligible probability that broadcast of $Tx_{\mathbb{RV},i}$ causes the honest party $\mathcal{A}$ to lose any funds in the channel.

Case 2. Let there exist an FPPW channel with n channel updates where $n \geq 0$. Assume that the honest channel party $\mathcal{P} \in \{\mathcal{A}, \mathcal{B}\}$ gets online when the last published block on the blockchain is BL_j. Party $\mathcal{P}$ observes that $Tx_{\mathbb{CM},i}$ with $i < n$ and $Tx_{\mathbb{RC}}$ have been published on-chain through the blocks BL_k and BL_l respectively with $k, l \leq j$ and $j+1 < l+T$.

Lemma 6. *If conditions of Case 2 are satisfied, the probability that $\mathcal{P}$ loses any funds in the channel is negligible.*

Proof. Without loss of generality let $\mathcal{P} = \mathcal{A}$. All subconditions of auxiliary output of $Tx_{\mathbb{CM},i}$ include either $pk_{\mathcal{A}}$ or $Y_{\mathcal{A},i}$. Thus, it is of negligible probability that any PPT adversary $\mathscr{A}$ spends the auxiliary output of $Tx_{\mathbb{CM},i}$ without $\mathcal{A}$'s authorization. Otherwise, the underlying digital signature or the hard relation would break. The channel party $\mathcal{A}$ grants such an authorization only on $Tx_{\mathbb{RV},i}$. However, based on Lemma 3, if $Tx_{\mathbb{RV},i}$ is published, the probability that $\mathcal{A}$ loses any funds is negligible. Thus, we assume that when $\mathcal{A}$ gets online, $Tx_{\mathbb{RV},i}$ is unpublished and hence $O^2_{\mathbb{CM},i}$ is unspent yet. According to values of j and k two categories of cases are possible. We firstly consider cases with $j+1 \geq k+t$ and prove that in such cases $\mathcal{A}$ can publish $Tx_{\mathbb{PN}_2,i}$. Then, we show that if $j+1 < k+t$, $\mathcal{A}$ can still publish $Tx_{\mathbb{RV},i}$. Thus, according to Lemma 3, for all cases, the probability that $\mathcal{A}$ loses any funds is negligible.

Consider cases with $j+1 \geq k+t$. The non-timelocked subcondition of $O_{\mathbb{RC}}$ includes $pk_{\mathcal{A}}$ and hence due to our assumption regarding the security of the underlying digital signature, it is of negligible probability that the output of $Tx_{\mathbb{RC}}$ is spent without $\mathcal{A}$'s authorization through the block BL_m with $m < l+T$. Also $\mathcal{A}$ grants this authorization only on $Tx_{\mathbb{PN}_2,i}$. Actually, all channel participants grant such an authorization on $Tx_{\mathbb{PN}_2,i}$. Additionally, since $Tx_{\mathbb{CM},i}$ is on-chain, the probability that $\mathcal{A}$ fails to obtain $y_{\mathcal{B},i}$ and hence fails to generate the required signatures for the first input of $Tx_{\mathbb{PN}_2,i}$ is negligible. Otherwise, $\mathsf{aEUF-CMA}$ security or extractability of the used adaptor signature is violated. Thus, party $\mathcal{A}$ might publish $Tx_{\mathbb{PN}_2,i}$ on-chain through one of the blocks $BL_{j+1}, \cdots, BL_{l+T-1}$ and since based on assumptions of Case 2 we have $j+1 \leq l+T-1$, the honest party $\mathcal{A}$ will have at least $l+T-1-j \geq 1$ rounds time to publish $Tx_{\mathbb{PN}_2,i}$ which is enough based on our assumptions regarding the value of confirmation delay.

Now let $j+1 < k+t$. We know that without having the honest party $\mathcal{A}$'s authorization, it is of negligible probability that any PPT adversary is able to spend the first or the second output of $Tx_{\mathbb{CM},i}$ through one of the blocks $BL_{k+1}, \cdots, BL_{k+t-1}$. All the channel participants grant such authorizations only on $Tx_{\mathbb{RV},i}$ and $\mathcal{A}$ has created this transaction in step 6 of the channel update phase from state i to $i+1$. Thus, $\mathcal{A}$ can publish it through one of the blocks $BL_{j+1}, \cdots, BL_{k+t-1}$ and since $j+1 < k+t$, $\mathcal{A}$ will have at least $k+t-1-j \geq 1$ rounds time to publish $Tx_{\mathbb{RV},i}$.

Therefore, for all cases, $\mathcal{A}$ can publish either $Tx_{\mathbb{RV},i}$ or $Tx_{\mathbb{PN}_2,i}$ and hence according to Lemma 3, the probability that $\mathcal{A}$ loses any funds is negligible.

Case 3. Let there exist an FPPW channel with n channel updates where $n \geq 0$. Assume that the honest channel party $\mathcal{P} \in \{\mathcal{A}, \mathcal{B}\}$ gets online when the last published block on the blockchain is BL_j. Party $\mathcal{P}$ observes that $Tx_{\mathbb{CM},i}$ has been published on-chain through the block BL_k with $k \leq j$ but $Tx_{\mathbb{RC}}$ is unpublished.

Lemma 7. *If conditions of Case 3 are satisfied, the probability that $\mathcal{P}$ loses any funds in the channel is negligible.*

Proof. Without loss of generality let $\mathcal{P} = \mathcal{A}$. Similar to the proof of Lemma 6, we assume that $Tx_{\mathbb{RV},i}$ is unpublished because otherwise the probability that $\mathcal{A}$ loses some funds is negligible. According to proof of Lemma 6, if we have $j+1 < k+t$, $\mathcal{A}$ will have at least $k+t-1-j \geq 1$ rounds time to publish $Tx_{\mathbb{RV},i}$.

Now let $j+1 \geq k+t$. We know that due to security of the underlying digital signature, it is of negligible probability that someone spend $O_{\mathbb{CL}}$ without $\mathcal{A}$'s authorization and $\mathcal{A}$ grants this authorization only on $Tx_{\mathbb{RC}}$ and $Tx_{\mathbb{PN}_1,i}$. Also, since $Tx_{\mathbb{CM},i}$ is on-chain, the probability that $\mathcal{A}$ fails to obtain $y_{\mathcal{B},i}$ is negligible. Thus, it is of negligible probability that party $\mathcal{A}$ cannot generate the required signatures for the first input of $Tx_{\mathbb{PN}_1,i}$. Then, party $\mathcal{A}$ can publish $Tx_{\mathbb{PN}_1,i}$ on-chain through one of the blocks $BL_{j+1}, BL_{j+2}, \cdots$. Based on Lemma 3, it is of negligible probability that broadcast of $Tx_{\mathbb{PN}_1,i}$ can cause $\mathcal{A}$ to lose any funds.

If before broadcast of $Tx_{\mathbb{PN}_1,i}$, $Tx_{\mathbb{RC}}$ is published by the watchtower, conditions of Case 2 get satisfied and due to Lemma 6, we know that it is of negligible probability that $\mathcal{A}$ loses any funds in the channel.

Proof of Lemma 1. Without loss of generality let $\mathcal{P} = \mathcal{A}$. We know that based on Lemma 4, $\mathcal{A}$ does not lose any funds with non-negligible probability unless a revoked commit transaction is published on-chain. However, at the end of channel establishment phase, no revoked commit transaction $Tx_{\mathbb{CM},i}$ exists to be published on-chain. Furthermore, the channel establishment completes once $Tx_{\mathbb{CL}}$ is recorded on-chain and since $Tx_{\mathbb{RC}}$ spends the output of $Tx_{\mathbb{CL}}$, right after the end of the channel establishment phase, neither a revoked commit transaction nor the reclaim transaction are on-chain. Now according to assumptions, $\mathcal{A}$ checks the chain at the end of the channel establishment phase and goes offline for at most $T-1$ rounds. The next time that $\mathcal{A}$ gets online, there will be 4 possibilities regarding broadcast of a revoked $Tx_{\mathbb{CM},i}$ or $Tx_{\mathbb{RC}}$ during the time interval when $\mathcal{A}$ has been offline:

1. Only $Tx_{\mathbb{RC}}$ has been published on-chain,
2. Both a revoked $Tx_{\mathbb{CM},i}$ and $Tx_{\mathbb{RC}}$ have been published,
3. Only a revoked $Tx_{\mathbb{CM},i}$ has been published, or
4. neither a revoked $Tx_{\mathbb{CM},i}$ nor $Tx_{\mathbb{RC}}$ has been published.

It can be seen that possibilities 1, 2, and 3 correspond with Case 1, Case 2, and Case 3 respectively and for these cases based on Lemmas 5, 6, and 7, the probability that $\mathcal{A}$ loses any funds is negligible. If neither a revoked commit transaction nor the reclaim transaction are on-chain (possibility 4) $\mathcal{A}$ can again go offline for at most $T-1$ rounds.

Lemma 8. *For an FPPW channel with n channel updates, an honest watchtower $\mathcal{W}$ does not lose any funds with non-negligible probability unless first a revoked commit transaction $Tx_{\mathbb{CM},i}$ with $i < n$ is broadcast on the chain and at least t rounds later, either $Tx_{\mathbb{PN}_1,i}$ or $Tx_{\mathbb{PN}_2,i}$ is also published on the chain.*

Proof. Assume that channel update phase has completed n times with $n \geq 0$. We discuss different scenarios using which the honest watchtower can be cheated. Funds of the watchtower are locked in $O_{\mathbb{CL}}$ and condition of $O_{\mathbb{CL}}$ includes $pk_{\mathcal{W}}$. Thus, due to security of the underlying digital signature, it is of negligible probability that any PPT adversary $\mathscr{A}$ spends $O_{\mathbb{CL}}$ without $\mathcal{W}$'s authorization. $Tx_{\mathbb{RC}}$ and $Tx_{\mathbb{PN}_1,i}$ with $i = [0, n-1]$ are the only transactions in the protocol that spend $O_{\mathbb{CL}}$ and $\mathcal{W}$ grants authorization for. Both subconditions of $O_{\mathbb{RC}}$ include $pk_{\mathcal{W}}$. Thus, due to security of the underlying digital signature, it is of negligible probability that any PPT adversary $\mathscr{A}$ spends $O_{\mathbb{RC}}$ without $\mathcal{W}$'s authorization and $Tx_{\mathbb{PN}_2,i}$ with $i = [0, n-1]$ are only transactions in the protocol that spend $O_{\mathbb{RC}}$ and $\mathcal{W}$ grants authorization for. Therefore, all scenarios with non-negligible probability lead to broadcast of $Tx_{\mathbb{PN}_1,i}$ or $Tx_{\mathbb{PN}_2,i}$. Since both $Tx_{\mathbb{PN}_1,i}$ and $Tx_{\mathbb{PN}_2,i}$ take the auxiliary output of $Tx_{\mathbb{CM},i}$ as their first inputs, those transactions can only be published if $Tx_{\mathbb{CM},i}$ is on-chain. Meeting the non-timelocked subcondition of $O^2_{\mathbb{CM},i}$ requires $\mathcal{W}$'s authorization and $\mathcal{W}$ does not grant such an authorization on $Tx_{\mathbb{PN}_1,i}$ or $Tx_{\mathbb{PN}_2,i}$. Therefore, it is of negligible probability that any PPT adversary publishes $Tx_{\mathbb{PN}_1,i}$ or $Tx_{\mathbb{PN}_2,i}$ before t rounds being elapsed since broadcast of $Tx_{\mathbb{CM},i}$.

Lemma 9. *For an FPPW channel with n channel updates, if $Tx_{\mathbb{CM},i}$ with $i \in [0, n-1]$ is published on-chain, it is of negligible probability that broadcast of $Tx_{\mathbb{RV},i}$ causes the honest watchtower $\mathcal{W}$ to lose any funds in the channel.*

Proof. Based on Lemma 8, the probability of cheating the watchtower without publishing $Tx_{\mathbb{PN}_1,i}$ or $Tx_{\mathbb{PN}_2,i}$ is negligible. However, since $Tx_{\mathbb{RV},i}$, $Tx_{\mathbb{PN}_1,i}$ and $Tx_{\mathbb{PN}_2,i}$ spend auxiliary output of $Tx_{\mathbb{CM},i}$, if $Tx_{\mathbb{RV},i}$ is published on-chain, $Tx_{\mathbb{PN}_1,i}$ and $Tx_{\mathbb{PN}_2,i}$ cannot be recorded on-chain anymore.

Proof of Lemma 2 Based on Lemma 8, $\mathcal{W}$ does not lose any funds with non-negligible probability unless a revoked commit transaction and then at least t rounds later its corresponding $Tx_{\mathbb{PN}_1,i}$ or $Tx_{\mathbb{PN}_2,i}$ are published. However, at the end of the channel establishment phase, no revoked commit transaction exists to be published yet. Also, following our assumptions, $\mathcal{W}$ remains online after the channel establishment. Now assume that a revoked $Tx_{\mathbb{CM},i}$ is published through the block BL_j. The timelocked subconditions of outputs of $Tx_{\mathbb{CM},i}$ cannot be met within $t-1$ rounds. Also, meeting the non-timelocked subconditions of $O^1_{\mathbb{CM},i}$ and $O^2_{\mathbb{CM},i}$ requires $\mathcal{W}$'s signature and $\mathcal{W}$ does not grant such authorizations on any transaction other than $Tx_{\mathbb{RV},i}$. Thus, once $Tx_{\mathbb{CM},i}$ is published on the chain, due to our assumption regarding the security of the underlying digital signature, it is of negligible probability that any adversary $\mathscr{A}$ spends $O^1_{\mathbb{CM},i}$ or $O^2_{\mathbb{CM},i}$ within $t-1$ rounds using any transaction other than $Tx_{\mathbb{RV},i}$. Furthermore, $\mathcal{W}$ has received $Tx_{\mathbb{RV},i}$ (through step 6 of the channel update phase from state i to $i+1$), before giving authorization on second input of $Tx_{\mathbb{PN}_1,i}$ or $Tx_{\mathbb{PN}_2,i}$ (through step 7 of the channel update phase from state i to $i+1$). Thus, once $Tx_{\mathbb{CM},i}$ is published, $\mathcal{W}$ can publish $Tx_{\mathbb{RV},i}$ through one of the blocks $BL_{j+1} \cdots BL_{j+t-1}$, meaning that the honest watchtower has at least $t-1$ rounds time to publish $Tx_{\mathbb{RV},i}$, which is enough based on our blockchain assumptions regarding the value of the confirmation delay. Following Lemma 9, it is of negligible probability that broadcast of $Tx_{\mathbb{RV},i}$ causes the honest watchtower $\mathcal{W}$ to lose any funds in the channel.

B A New Variant of Generalized Channel

A generalized channel [1] consists of three transaction types including funding, commit and split transactions. Fig. 5 illustrates the transaction flows for a generalized channel and the way the channel state is updated from state i to $i+1$. The public/private key pair $(R_{\mathcal{P},i}, r_{\mathcal{P},i})$ with $\mathcal{P} \in \{\mathcal{A}, \mathcal{B}\}$ is the revocation key pair generated by $\mathcal{P}$ for the i^{th} state. As is clear, the channel is updated in two sub-phases. In the first sub-phase, the new commit and split transactions are created and signed by parties. Party $\mathcal{A}$ ($\mathcal{B}$) uses the adaptor signature and its counter-party's statement $Y_{\mathcal{B},i+1}$ ($Y_{\mathcal{A},i+1}$) to sign the commit transaction $Tx_{\mathbb{CM},i+1}$. In the second sub-phase, revocation keys of state i $(r_{\mathcal{A},i}, r_{\mathcal{B},i})$ are exchanged between the parties.

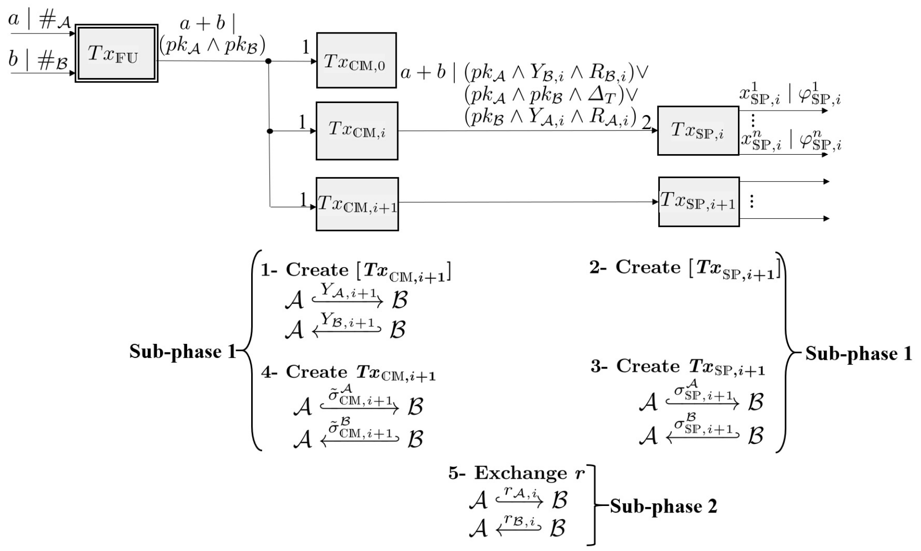
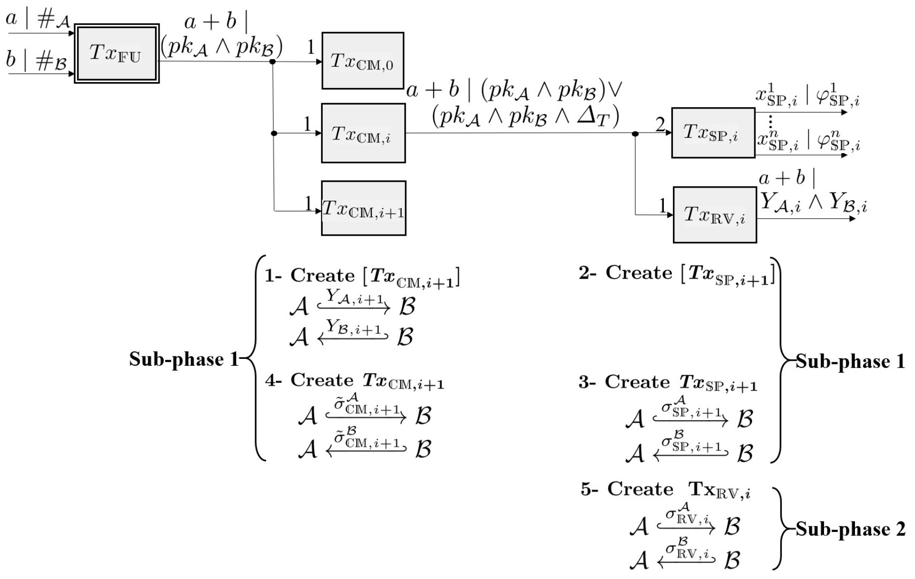

$a \mid \#_{\mathcal{A}}$
$b \mid \#_{\mathcal{B}}$
$Tx_{\mathbb{FU}}$
$a+b \mid (pk_{\mathcal{A}} \wedge pk_{\mathcal{B}})$
1
$Tx_{\mathbb{CM},0}$
$Tx_{\mathbb{CM},i}$
$Tx_{\mathbb{CM},i+1}$
$a+b \mid (pk_{\mathcal{A}} \wedge Y_{\mathcal{B},i} \wedge R_{\mathcal{B},i}) \vee (pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge \Delta_T) \vee (pk_{\mathcal{B}} \wedge Y_{\mathcal{A},i} \wedge R_{\mathcal{A},i})$
2
$Tx_{\mathbb{SP},i}$
$Tx_{\mathbb{SP},i+1}$
$x^1_{\mathbb{SP},i} \mid \varphi^1_{\mathbb{SP},i}$
$x^n_{\mathbb{SP},i} \mid \varphi^n_{\mathbb{SP},i}$
1- Create $[Tx_{\mathbb{CM},i+1}]$
$\mathcal{A} \xrightarrow{Y_{\mathcal{A},i+1}} \mathcal{B}$
$\mathcal{A} \xleftarrow{Y_{\mathcal{B},i+1}} \mathcal{B}$
2- Create $[Tx_{\mathbb{SP},i+1}]$
3- Create $Tx_{\mathbb{SP},i+1}$
$\mathcal{A} \xrightarrow{\sigma^{\mathcal{A}}_{\mathbb{SP},i+1}} \mathcal{B}$
$\mathcal{A} \xleftarrow{\sigma^{\mathcal{B}}_{\mathbb{SP},i+1}} \mathcal{B}$
4- Create $Tx_{\mathbb{CM},i+1}$
$\mathcal{A} \xrightarrow{\tilde{\sigma}^{\mathcal{A}}_{\mathbb{CM},i+1}} \mathcal{B}$
$\mathcal{A} \xleftarrow{\tilde{\sigma}^{\mathcal{B}}_{\mathbb{CM},i+1}} \mathcal{B}$
Sub-phase 1
Sub-phase 1
5- Exchange r
$\mathcal{A} \xrightarrow{r_{\mathcal{A},i}} \mathcal{B}$
$\mathcal{A} \xleftarrow{r_{\mathcal{B},i}} \mathcal{B}$
Sub-phase 2

Fig. 5. A Generalized Channel

Now assume that a party, let's say party $\mathcal{A}$, broadcasts the revoked commit transaction $Tx_{\mathbb{CM},i}$ on-chain. Since $Tx_{\mathbb{CM},i}$ spends the output of $Tx_{\mathbb{FU}}$, it must include signatures of both parties. However, Party $\mathcal{B}$ has provided one pre-signature for this transaction to party $\mathcal{A}$, which is $\tilde{\sigma}^{\mathcal{B}}_{\mathbb{CM},i}$. Thus, the only way that party $\mathcal{A}$ is able to produce the valid signature of party $\mathcal{B}$ on $[Tx_{\mathbb{CM},i}]$ is adapting the pre-signature. However, it reveals the value of the witness $y_{\mathcal{A},i}$ to $\mathcal{B}$. Then, party $\mathcal{B}$ will be able to use the third subcondition of $O_{\mathbb{CM},i}$ and spend it with no timelock. The first subcondition cannot be used because only party $\mathcal{B}$ knows the value of $y_{\mathcal{B},i}$. The second subcondition is also timelocked and is used by the split transaction.

Now we change the introduced scheme such that rather than exchanging the revocation keys, the old commit transaction is revoked by exchanging signatures on the corresponding revocation transaction. Transaction flows and channel update phase for the new variant of generalized channel are depicted in Fig. 6. As is clear, if the revoked commit transaction $Tx_{\mathbb{CM},i}$ is published by a channel party, let's say $\mathcal{A}$, the corresponding split transaction cannot be broadcast within $t-1$ rounds. However, $\mathcal{B}$ can immediately publish the revocation transaction $Tx_{\mathrm{RV},i}$. Then, since only $\mathcal{B}$ knows both $y_{\mathcal{A},i}$ and $y_{\mathcal{B},i}$, $\mathcal{B}$ will be actually the owner of all the channel funds. The new variant of the generalized channel could be useful for some applications. Actually, FPPW channel is an extension of this new variant.



$a \mid \#_{\mathcal{A}}$
$b \mid \#_{\mathcal{B}}$
Tx_{FU}
$a+b \mid (pk_{\mathcal{A}} \wedge pk_{\mathcal{B}})$
1
$Tx_{\mathrm{CM},0}$
1
$Tx_{\mathrm{CM},i}$
1
$Tx_{\mathrm{CM},i+1}$
$a+b \mid (pk_{\mathcal{A}} \wedge pk_{\mathcal{B}}) \vee (pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge \Delta_T)$
2
$Tx_{\mathrm{SP},i}$
$x^1_{\mathrm{SP},i} \mid \varphi^1_{\mathrm{SP},i}$
$x^n_{\mathrm{SP},i} \mid \varphi^n_{\mathrm{SP},i}$
1
$Tx_{\mathrm{RV},i}$
$a+b \mid Y_{\mathcal{A},i} \wedge Y_{\mathcal{B},i}$
1- Create $[\boldsymbol{Tx}_{\mathrm{CM},i+1}]$
$\mathcal{A} \xrightarrow{Y_{\mathcal{A},i+1}} \mathcal{B}$
$\mathcal{A} \xleftarrow{Y_{\mathcal{B},i+1}} \mathcal{B}$
Sub-phase 1
4- Create $\boldsymbol{Tx}_{\mathrm{CM},i+1}$
$\mathcal{A} \xrightarrow{\tilde{\sigma}^{\mathcal{A}}_{\mathrm{CM},i+1}} \mathcal{B}$
$\mathcal{A} \xleftarrow{\tilde{\sigma}^{\mathcal{B}}_{\mathrm{CM},i+1}} \mathcal{B}$
2- Create $[\boldsymbol{Tx}_{\mathrm{SP},i+1}]$
3- Create $\boldsymbol{Tx}_{\mathrm{SP},i+1}$
$\mathcal{A} \xrightarrow{\sigma^{\mathcal{A}}_{\mathrm{SP},i+1}} \mathcal{B}$
$\mathcal{A} \xleftarrow{\sigma^{\mathcal{B}}_{\mathrm{SP},i+1}} \mathcal{B}$
Sub-phase 1
5- Create $\mathbf{Tx}_{\mathrm{RV},i}$
$\mathcal{A} \xrightarrow{\sigma^{\mathcal{A}}_{\mathrm{RV},i}} \mathcal{B}$
$\mathcal{A} \xleftarrow{\sigma^{\mathcal{B}}_{\mathrm{RV},i}} \mathcal{B}$
Sub-phase 2

Fig. 6. A New Variant of Generalized Bitcoin Channel

C FPPW Protocol

In this section, protocols for different phases of FPPW will be presented. In different steps of the protocol, channel participants generate (or verify) some signatures or pre-signatures on protocol transactions. When a signature or pre-signature is going to be generated (or verified) for j^{th} input of the transaction Tx_i, the input message to the signing (or verification) algorithm is denoted by $f([Tx_i], j)$ [11].

Remark 5. The FPPW protocol which is presented in this Appendix is slightly different from what was explained in Section 4. Those differences are as following. In the channel establishment phase, $[Tx_{\mathrm{RV},0}]$ is also created and $\mathcal{W}$'s signatures on this transaction are given to both $\mathcal{A}$ and $\mathcal{B}$. This transaction lacks channel parties' signatures. Those signatures are computed in the channel update phase when $Tx_{\mathrm{CM},0}$ is going to be revoked. However, $\mathcal{A}$ and $\mathcal{B}$ require $\mathcal{W}$'s signature in this phase to be able to continue using the channel given that $\mathcal{W}$ will be non-cooperative or temporarily unavailable in the channel update phase from state 0 to 1. Otherwise, $\mathcal{A}$ and $\mathcal{B}$ will have to update the channel on-chain because $Tx_{\mathbb{CM},0}$ will be irrevocable. Similarly, in the update channel phase from state i to $i+1$, $\mathcal{W}$ also generates its signatures for both inputs of $Tx_{\mathbb{RV},i+1}$ and sends them to $\mathcal{A}$ and $\mathcal{B}$. This will allow $\mathcal{A}$ and $\mathcal{B}$ to update the channel from state $i+1$ to $i+2$ and revoke $Tx_{\mathbb{CM},i+1}$ even if $\mathcal{W}$ will be temporarily unavailable or uncooperative. Otherwise, $Tx_{\mathbb{CM},i+1}$ will be irrevocable.

FPPW channel establishment protocol is as following:

Preconditions: $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{W}$ own $a+\epsilon/2$, $b+\epsilon/2$ and $c=a+b$ coins on-chain in output of transactions with transaction identifiers $txid_{\mathcal{A}}$, $txid_{\mathcal{B}}$ and $txid_{\mathcal{W}}$ respectively. $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{W}$ know each other's public keys, $pk_{\mathcal{A}}$, $pk_{\mathcal{B}}$ and $pk_{\mathcal{W}}$ and values of ϵ, a, b and c that are going to be used in the channel.

1. Create $[Tx_{\mathbb{FU}}]$:
   (a) $\mathcal{P} \in \{\mathcal{A},\mathcal{B}\} \xhookrightarrow{txid_{\mathcal{P}}} \bar{\mathcal{P}}, \mathcal{W}$
   (b) If $\mathcal{P}$ receives $txid_{\bar{\mathcal{P}}}$, it creates $[Tx_{\mathbb{FU}}]$ according to 1. Else it stops.
   (c) If $\mathcal{W}$ receives $txid_{\mathcal{P}}$ with $\mathcal{P}=\{\mathcal{A},\mathcal{B}\}$, it creates $[Tx_{\mathbb{FU}}]$ according to 1. Else it stops.
2. Create $[Tx_{\mathbb{CM},0}]$:
   (a) $\mathcal{P} \in \{\mathcal{A},\mathcal{B}\}$ generates $(Y_{\mathcal{P},0}, y_{\mathcal{P},0}) \leftarrow \mathsf{GenR}$.
   (b) $\mathcal{P} \xhookrightarrow{Y_{\mathcal{P},0}} \bar{\mathcal{P}}, \mathcal{W}$
   (c) If $\mathcal{P}$ receives $Y_{\bar{\mathcal{P}},0}$, it creates $[Tx_{\mathbb{CM},0}]$ according to 2. Else it stops.
   (d) If $\mathcal{W}$ receives $Y_{\mathcal{P},0}$ with $\mathcal{P}=\{\mathcal{A},\mathcal{B}\}$, it creates $[Tx_{\mathbb{CM},0}]$ according to 2. Else it stops.
3. Create $[Tx_{\mathbb{SP},0}]$: Party $\mathcal{P} \in \{\mathcal{A},\mathcal{B}\}$ creates $[Tx_{\mathbb{SP},0}]$ according to 3.
4. Create $[Tx_{\mathbb{RV},0}]$:
   (a) $\mathcal{W}$ creates $[Tx_{\mathbb{RV},0}]$ according to 6.
   (b) $\mathcal{W}$ computes $\sigma^{\mathcal{W},j}_{\mathbb{RV},0} = \mathsf{Sign}_{sk_{\mathcal{W}}}(f([Tx_{\mathbb{RV},0}], j))$ with $j=\{1,2\}$.
   (c) $\mathcal{W} \xhookrightarrow{\sigma^{\mathcal{W},1}_{\mathbb{RV},0}, \sigma^{\mathcal{W},2}_{\mathbb{RV},0}} \mathcal{A}, \mathcal{B}$.
   (d) If party $\mathcal{P} \in \{\mathcal{A},\mathcal{B}\}$ receives $\sigma^{\mathcal{W},1}_{\mathbb{RV},0}$ and $\sigma^{\mathcal{W},2}_{\mathbb{RV},0}$ from $\mathcal{W}$ s.t. $\mathsf{Vrfy}_{pk_{\mathcal{W}}}(f([Tx_{\mathbb{RV},0}], j); \sigma^{\mathcal{W},j}_{\mathbb{RV},0}) = 1$ with $j=\{1,2\}$, it continues. Else it stops.
5. Create $Tx_{\mathbb{SP},0}$:
   (a) Party $\mathcal{P} \in \{\mathcal{A},\mathcal{B}\}$ computes $\sigma^{\mathcal{P}}_{\mathbb{SP},0} = \mathsf{Sign}_{sk_{\mathcal{P}}}(f([Tx_{\mathbb{SP},0}], 1))$.
   (b) $\mathcal{P} \xhookrightarrow{\sigma^{\mathcal{P}}_{\mathbb{SP},0}} \bar{\mathcal{P}}$
   (c) If party $\mathcal{P}$ receives $\sigma^{\bar{\mathcal{P}}}_{\mathbb{SP},0}$ s.t. $\mathsf{Vrfy}_{pk_{\bar{\mathcal{P}}}}(f([Tx_{\mathbb{SP},0}], 1); \sigma^{\bar{\mathcal{P}}}_{\mathbb{SP},0}) = 1$, it continues. Else it stops.
   (d) Party $\mathcal{P}$ creates $Tx_{\mathbb{SP},0}$.
6. Create $Tx_{\mathbb{CM},0}$:

   (a) Party $\mathcal{P} \in \{\mathcal{A}, \mathcal{B}\}$ computes $\tilde{\sigma}^{\mathcal{P}}_{\mathbb{CM},0} = \mathsf{pSign}_{sk_{\mathcal{P}}}(f([Tx_{\mathbb{CM},0}], 1), Y_{\bar{\mathcal{P}},0})$.
   (b) $\mathcal{P} \xhookrightarrow{\tilde{\sigma}^{\mathcal{P}}_{\mathbb{CM},0}} \bar{\mathcal{P}}$
   (c) If party $\mathcal{P}$ receives $\tilde{\sigma}^{\bar{\mathcal{P}}}_{\mathbb{CM},0}$ s.t. $\mathsf{pVrfy}_{pk_{\bar{\mathcal{P}}}}(f([Tx_{\mathbb{CM},0}], 1), Y_{\mathcal{P},0}; \tilde{\sigma}^{\bar{\mathcal{P}}}_{\mathbb{CM},0}) = 1$, it computes $\sigma^{\bar{\mathcal{P}}}_{\mathbb{CM},0} = \mathsf{Adapt}(\tilde{\sigma}^{\bar{\mathcal{P}}}_{\mathbb{CM},0}, y_{\mathcal{P},0})$, computes $\sigma^{\mathcal{P}}_{\mathbb{CM},0} = \mathsf{Sign}_{sk_{\mathcal{P}}}$ $(f([Tx_{\mathbb{CM},0}], 1))$, creates $Tx_{\mathbb{CM},0}$ and continues. Else it stops.
7. Create $Tx_{\mathbb{FU}}$:
   (a) Party $\mathcal{P} \in \{\mathcal{A}, \mathcal{B}\}$ computes $\sigma^{\mathcal{P}}_{\mathbb{FU}} = \mathsf{Sign}_{sk_{\mathcal{P}}}(f([Tx_{\mathbb{FU}}], j))$ where $j := 1$ if $\mathcal{P} = \mathcal{A}$ or $j := 2$ otherwise.
   (b) $\mathcal{P} \xhookrightarrow{\sigma^{\mathcal{P}}_{\mathbb{FU}}} \bar{\mathcal{P}}$
   (c) If party $\mathcal{P}$ receives $\sigma^{\bar{\mathcal{P}}}_{\mathbb{FU}}$ s.t. $\mathsf{Vrfy}_{pk_{\bar{\mathcal{P}}}}(f([Tx_{\mathbb{FU}}], j); \sigma^{\bar{\mathcal{P}}}_{\mathbb{FU}}) = 1$ with $j := 1$ if $\mathcal{P} = \mathcal{B}$ or $j := 2$ otherwise, it continues. Else it stops.
   (d) Party $\mathcal{P}$ creates $Tx_{\mathbb{FU}}$.
8. Publish $Tx_{\mathbb{FU}}$: Party $\mathcal{P}$ publishes $Tx_{\mathbb{FU}}$ on-chain.
9. Create $[Tx_{\mathbb{CL}}]$:
   (a) $\mathcal{W}$ creates $[Tx_{\mathbb{CL}}]$ according to 4.
   (b) $\mathcal{W} \xhookrightarrow{txid_{\mathcal{W}}} \mathcal{A}, \mathcal{B}$.
   (c) If party $\mathcal{P}$ receives $txid_{\mathcal{W}}$, it creates $[Tx_{\mathbb{CL}}]$. Else it stops.
10. Create $[Tx_{\mathbb{RC}}]$: $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{W}$ creates $[Tx_{\mathbb{RC}}]$ according to 5.
11. Create $Tx_{\mathbb{RC}}$:
   (a) Party $\mathcal{P} \in \{\mathcal{A}, \mathcal{B}\}$ computes $\sigma^{\mathcal{P}}_{\mathbb{RC}} = \mathsf{Sign}_{sk_{\mathcal{P}}}(f([Tx_{\mathbb{RC}}], 1))$.
   (b) $\mathcal{P} \xhookrightarrow{\sigma^{\mathcal{P}}_{\mathbb{RC}}} \mathcal{W}$
   (c) If $\mathcal{W}$ receives $\sigma^{\mathcal{P}}_{\mathbb{RC}}$ with $\mathcal{P} = \{\mathcal{A}, \mathcal{B}\}$ s.t. $\mathsf{Vrfy}_{pk_{\mathcal{P}}}(f([Tx_{\mathbb{RC}}], 1); \sigma^{\mathcal{P}}_{\mathbb{RC}}) = 1$, it continues. Else it stops.
   (d) $\mathcal{W}$ computes $\sigma^{\mathcal{W}}_{\mathbb{RC}} = \mathsf{Sign}_{sk_{\mathcal{W}}}(f([Tx_{\mathbb{RC}}], 1))$ and creates $Tx_{\mathbb{RC}}$.
12. Create $Tx_{\mathbb{CL}}$:
   (a) $\mathcal{W}$ computes $\sigma^{\mathcal{W}}_{\mathbb{CL}} = \mathsf{Sign}_{sk_{\mathcal{W}}}(f([Tx_{\mathbb{CL}}], 1))$.
   (b) $\mathcal{W}$ creates $Tx_{\mathbb{CL}}$.
13. Publish $Tx_{\mathbb{CL}}$: $\mathcal{W}$ publishes $Tx_{\mathbb{CL}}$ on-chain.

FPPW channel update protocol is as following:

Preconditions: The channel establishment phase is complete and $Tx_{\mathbb{FU}}$ and $Tx_{\mathbb{CL}}$ are on-chain. The channel update phase has completed i times and hence the channel is at state i.

1. Create $[Tx_{\mathbb{CM},i+1}]$:
   (a) Party $\mathcal{P} \in \{\mathcal{A}, \mathcal{B}\}$ generates $(Y_{\mathcal{P},i+1}, y_{\mathcal{P},i+1}) \leftarrow \mathsf{GenR}$.
   (b) $\mathcal{P} \xhookrightarrow{Y_{\mathcal{P},i+1}} \bar{\mathcal{P}}, \mathcal{W}$
   (c) If Party $\mathcal{P}$ receives $Y_{\bar{\mathcal{P}},i+1}$, it creates $[Tx_{\mathbb{CM},i+1}]$ according to 2. Else it stops.

   (d) If $\mathcal{W}$ receives $Y_{\mathcal{P},i+1}$ with $\mathcal{P} = \{\mathcal{A}, \mathcal{B}\}$, it creates $[Tx_{\mathbb{CM},i+1}]$ according to 2. Else it stops.
2. Create $[Tx_{\mathbb{SP},i+1}]$: Party $\mathcal{P} \in \{\mathcal{A}, \mathcal{B}\}$ creates $[Tx_{\mathbb{SP},i+1}]$ according to 3.
3. Create $[Tx_{\mathbb{RV},i+1}]$:
   (a) $\mathcal{W}$ creates $[Tx_{\mathbb{RV},i+1}]$ according to 6.
   (b) $\mathcal{W}$ computes $\sigma^{\mathcal{W},j}_{\mathbb{RV},i+1} = \mathsf{Sign}_{sk_{\mathcal{W}}}([f(Tx_{\mathbb{RV},i+1}], j))$ with $j = \{1, 2\}$.
   (c) $\mathcal{W} \xhookrightarrow{\sigma^{\mathcal{W},1}_{\mathbb{RV},i+1}, \sigma^{\mathcal{W},2}_{\mathbb{RV},i+1}} \mathcal{A}, \mathcal{B}$.
   (d) If party $\mathcal{P} \in \{\mathcal{A}, \mathcal{B}\}$ receives $\sigma^{\mathcal{W},1}_{\mathbb{RV},i+1}$ and $\sigma^{\mathcal{W},2}_{\mathbb{RV},i+1}$ from $\mathcal{W}$ s.t. $\mathsf{Vrfy}_{pk_{\mathcal{W}}}(f([Tx_{\mathbb{RV},i+1}], j); \sigma^{\mathcal{W},j}_{\mathbb{RV},i+1}) = 1$ with $j = \{1, 2\}$, it continues. Else it stops.
4. Create $Tx_{\mathbb{SP},i+1}$:
   (a) Party $\mathcal{P} \in \{\mathcal{A}, \mathcal{B}\}$ computes $\sigma^{\mathcal{P}}_{\mathbb{SP},i+1} = \mathsf{Sign}_{sk_{\mathcal{P}}}(f([Tx_{\mathbb{SP},i+1}], 1))$.
   (b) $\mathcal{P} \xhookrightarrow{\sigma^{\mathcal{P}}_{\mathbb{SP},i+1}} \bar{\mathcal{P}}$
   (c) If party $\mathcal{P}$ receives $\sigma^{\bar{\mathcal{P}}}_{\mathbb{SP},i+1}$ s.t. $\mathsf{Vrfy}_{pk_{\bar{\mathcal{P}}}}(f([Tx_{\mathbb{SP},i+1}], 1); \sigma^{\bar{\mathcal{P}}}_{\mathbb{SP},i+1}) = 1$, it continues. Else it stops.
   (d) Party $\mathcal{P}$ creates $Tx_{\mathbb{SP},i+1}$.
5. Create $Tx_{\mathbb{CM},i+1}$:
   (a) Party $\mathcal{P} \in \{\mathcal{A}, \mathcal{B}\}$ computes $\tilde{\sigma}^{\mathcal{P}}_{\mathbb{CM},i+1} = \mathsf{pSign}_{sk_{\mathcal{P}}}(f([Tx_{\mathbb{CM},i+1}], 1), Y_{\bar{\mathcal{P}},i+1})$.
   (b) $\mathcal{P} \xhookrightarrow{\tilde{\sigma}^{\mathcal{P}}_{\mathbb{CM},i+1}} \bar{\mathcal{P}}$
   (c) If party $\mathcal{P}$ receives $\tilde{\sigma}^{\bar{\mathcal{P}}}_{\mathbb{CM},i+1}$ s.t. $\mathsf{pVrfy}_{pk_{\bar{\mathcal{P}}}}(f([Tx_{\mathbb{CM},i+1}], 1), Y_{\mathcal{P},i+1}; \tilde{\sigma}^{\bar{\mathcal{P}}}_{\mathbb{CM},i+1}) = 1$, it computes $\sigma^{\bar{\mathcal{P}}}_{\mathbb{CM},i+1} = \mathsf{Adapt}(\tilde{\sigma}^{\bar{\mathcal{P}}}_{\mathbb{CM},i+1}, y_{\mathcal{P},i+1})$, computes $\sigma^{\mathcal{P}}_{\mathbb{CM},i+1} = \mathsf{Sign}_{sk_{\mathcal{P}}}$ $(f([Tx_{\mathbb{CM},i+1}], 1))$, creates $Tx_{\mathbb{CM},i+1}$ and continues. Else it execute the non-collaborative closure phase (from $\mathcal{P}$'s point of view the channel is still at state i).
6. Create $Tx_{\mathbb{RV},i}$:
   (a) Party $\mathcal{P} \in \{\mathcal{A}, \mathcal{B}\}$ creates $[Tx_{\mathbb{RV},i}]$ according to 6.
   (b) Party $\mathcal{P}$ computes $\sigma^{\mathcal{P},1}_{\mathbb{RV},i} = \mathsf{Sign}_{sk_{\mathcal{P}}}(f([Tx_{\mathbb{RV},i}], 1))$ as part of $\gamma^{1}_{\mathbb{RV},i}$ and $\sigma^{\mathcal{P},2}_{\mathbb{RV},i} = \mathsf{Sign}_{sk_{\mathcal{P}}}(f([Tx_{\mathbb{RV},i}], 2))$ as part of $\gamma^{2}_{\mathbb{RV},i}$.
   (c) $\mathcal{P} \xhookrightarrow{\sigma^{\mathcal{P},1}_{\mathbb{RV},i}, \sigma^{\mathcal{P},2}_{\mathbb{RV},i}} \bar{\mathcal{P}}$
   (d) If party $\mathcal{P}$ receives $\sigma^{\bar{\mathcal{P}},1}_{\mathbb{RV},i}$ and $\sigma^{\bar{\mathcal{P}},2}_{\mathbb{RV},i}$ from $\bar{\mathcal{P}}$ s.t. $\mathsf{Vrfy}_{pk_{\bar{\mathcal{P}}}}(f([Tx_{\mathbb{RV},i}]$ $, 1); \sigma^{\bar{\mathcal{P}},1}_{\mathbb{RV},i}) = 1$ and $\mathsf{Vrfy}_{pk_{\bar{\mathcal{P}}}}(f([Tx_{\mathbb{RV},i}], 2); \sigma^{\bar{\mathcal{P}},2}_{\mathbb{RV},i}) = 1$, it continues. Else it executes the non-collaborative closure phase (The channel now is at state $i + 1$).
   (e) $\mathcal{P} \xhookrightarrow{\sigma^{\mathcal{P},1}_{\mathbb{RV},i}, \sigma^{\mathcal{P},2}_{\mathbb{RV},i}} \mathcal{W}$

   (f) If $\mathcal{W}$ receives $\sigma^{\mathcal{P},1}_{\mathbb{RV},i}$ and $\sigma^{\mathcal{P},2}_{\mathbb{RV},i}$ from $\mathcal{P} = \{\mathcal{A},\mathcal{B}\}$ s.t. $\mathsf{Vrfy}_{pk_{\mathcal{P}}}(f([Tx_{\mathbb{RV},i}],1);\sigma^{\mathcal{P},1}_{\mathbb{RV},i}) = 1$ and $\mathsf{Vrfy}_{pk_{\mathcal{P}}}(f([Tx_{\mathbb{RV},i}],2);\sigma^{\mathcal{P},2}_{\mathbb{RV},i}) = 1$, it continues. Else it stops.
   (g) $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{W}$ create $Tx_{\mathbb{RV},i}$ using $[Tx_{\mathbb{RV},i}]$ and $\gamma^{j}_{\mathbb{RV},i} = (2,(\sigma^{\mathcal{A},j}_{\mathbb{RV},i},\sigma^{\mathcal{B},j}_{\mathbb{RV},i},\sigma^{\mathcal{W},j}_{\mathbb{RV},i}))$ with $j=\{1,2\}$.
7. Create $[Tx_{\mathbb{PN}_1,i}]$ and $[Tx_{\mathbb{PN}_2,i}]$
   (a) Party $\mathcal{P} \in \{\mathcal{A},\mathcal{B}\}$ creates $[Tx_{\mathbb{PN}_1,i}]$ and $[Tx_{\mathbb{PN}_2,i}]$ according to 7 and 8.
   (b) Party $\mathcal{P}$ computes $\sigma^{\mathcal{P},2}_{\mathbb{PN}_1,i} = \mathsf{Sign}_{sk_{\mathcal{P}}}(f([Tx_{\mathbb{PN}_1,i}],2))$ and $\sigma^{\mathcal{P},2}_{\mathbb{PN}_2,i} = \mathsf{Sign}_{sk_{\mathcal{P}}}(f([Tx_{\mathbb{PN}_2,i}],2))$ as part of $\gamma^{2}_{\mathbb{PN}_1,i}$ and $\gamma^{2}_{\mathbb{PN}_2,i}$ respectively.
   (c) $\mathcal{P} \xhookrightarrow{\sigma^{\mathcal{P},2}_{\mathbb{PN}_1,i},\sigma^{\mathcal{P},2}_{\mathbb{PN}_2,i}} \bar{\mathcal{P}}$
   (d) If $\mathcal{P}$ receives signatures $\sigma^{\bar{\mathcal{P}},2}_{\mathbb{PN}_1,i}$ and $\sigma^{\bar{\mathcal{P}},2}_{\mathbb{PN}_2,i}$ s.t. $\mathsf{Vrfy}_{pk_{\bar{\mathcal{P}}}}(f([Tx_{\mathbb{PN}_1,i}],2);\sigma^{\bar{\mathcal{P}},2}_{\mathbb{PN}_1,i}) = 1$ and $\mathsf{Vrfy}_{pk_{\bar{\mathcal{P}}}}(f([Tx_{\mathbb{PN}_2,i}],2);\sigma^{\bar{\mathcal{P}},2}_{\mathbb{PN}_2,i}) = 1$, it continues. Else it executes the non-collaborative closure phase (The channel now is at state $i+1$) or gets online at least once every $t-1$ blocks.
   (e) $\mathcal{W}$ creates $[Tx_{\mathbb{PN}_1,i}]$ and $[Tx_{\mathbb{PN}_2,i}]$ according to 7 and 8
   (f) $\mathcal{W}$ computes $\sigma^{\mathcal{W},2}_{\mathbb{PN}_1,i} = \mathsf{Sign}_{sk_{\mathcal{W}}}(f([Tx_{\mathbb{PN}_1,i}],2)$ and $\sigma^{\mathcal{W},2}_{\mathbb{PN}_2,i} = \mathsf{Sign}_{sk_{\mathcal{W}}}(f([Tx_{\mathbb{PN}_2,i}],2)$.
   (g) $W \xhookrightarrow{\sigma^{\mathcal{W},2}_{\mathbb{PN}_1,i},\sigma^{\mathcal{W},2}_{\mathbb{PN}_2,i}} \mathcal{P}$ with $\mathcal{P} = \{\mathcal{A},\mathcal{B}\}$.
   (h) If $\mathcal{P} \in \{\mathcal{A},\mathcal{B}\}$ receives signatures $\sigma^{\mathcal{W},2}_{\mathbb{PN}_1,i}$ and $\sigma^{\mathcal{W},2}_{\mathbb{PN}_2,i}$ from $\mathcal{W}$ s.t. $\mathsf{Vrfy}_{pk_{\mathcal{W}}}(f([Tx_{\mathbb{PN}_1,i}],2);\sigma^{\mathcal{W},2}_{\mathbb{PN}_1,i}) = 1$ and $\mathsf{Vrfy}_{pk_{\mathcal{W}}}(f([Tx_{\mathbb{PN}_2,i}],2);\sigma^{\mathcal{W},2}_{\mathbb{PN}_2,i}) = 1$ it continues. Else it executes the non-collaborative closure phase (The channel now is at state $i+1$) or gets online at least once every $t-1$ blocks.

FPPW channel collaborative closure protocol is as following:

Preconditions: The channel establishment phase is complete and $Tx_{\mathbb{FU}}$ and $Tx_{\mathbb{CL}}$ are on-chain. The channel is at state n.

1. Create $Tx_{\overline{\mathbb{SP}}}$:
   (a) Party $\mathcal{P} \in \{\mathcal{A},\mathcal{B}\}$ creates $[Tx_{\overline{\mathbb{SP}}}]$ according to 9.
   (b) $\mathcal{P}$ computes $\sigma^{\mathcal{P}}_{\overline{\mathbb{SP}}} = \mathsf{Sign}_{sk_{\mathcal{P}}}(f([Tx_{\overline{\mathbb{SP}}}],1))$.
   (c) $\mathcal{P} \xhookrightarrow{\sigma^{\mathcal{P}}_{\overline{\mathbb{SP}}}} \bar{\mathcal{P}}$
   (d) If $\mathcal{P}$ receives $\sigma^{\bar{\mathcal{P}}}_{\overline{\mathbb{SP}}}$ from $\bar{\mathcal{P}}$ s.t. $\mathsf{Vrfy}_{pk_{\bar{\mathcal{P}}}}(f([Tx_{\overline{\mathbb{SP}}}],1);\sigma^{\bar{\mathcal{P}}}_{\overline{\mathbb{SP}}}) = 1$, it continues. Else it executes the non-collaborative closure phase (from $\mathcal{P}$'s point of view the channel is still at state n).
2. Publish $Tx_{\overline{\mathbb{SP}}}$: Party $\mathcal{P} \in \{\mathcal{A},\mathcal{B}\}$ publishes $Tx_{\overline{\mathbb{SP}}}$ on-chain.

FPPW channel non-collaborative closure protocol is as following:

Preconditions: The channel establishment phase is complete and $Tx_{\mathbb{FU}}$ and $Tx_{\mathbb{CL}}$ are on-chain. The channel is at state n.

1. Party $\mathcal{P} \in \{\mathcal{A}, \mathcal{B}\}$ publishes $Tx_{\mathbb{CM},n}$ on-chain.
2. Once $Tx_{\mathbb{CM},n}$ is recorded on-chain, $\mathcal{P}$ waits for t rounds and then publishes $Tx_{\mathbb{SP},n}$ on-chain.

The protocol for penalizing the cheating party is as following:

preconditions: The channel establishment phase is complete and $Tx_{\mathbb{FU}}$ and $Tx_{\mathbb{CL}}$ are on-chain. The channel is at state n. $Tx_{\mathbb{CM},i}$ with $i < n$ is recorded on-chain by a channel party. The watchtower is always online.

1. $\mathcal{W}$ observes that $Tx_{\mathbb{CM},i}$ is on-chain.
2. $\mathcal{W}$ publishes $Tx_{\mathbb{RV},i}$ on-chain before t rounds being elapsed since broadcast of $Tx_{\mathbb{CM},i}$.

The protocol for penalizing the unresponsive watchtower is as following:

preconditions: The channel establishment phase is complete and $Tx_{\mathbb{FU}}$ and $Tx_{\mathbb{CL}}$ are on-chain. The channel update phase has successfully completed n times. $Tx_{\mathbb{CM},i}$ with $i < n$ is recorded on-chain. Parties check the blockchain at least once every $T - 1$ rounds.

1. Party $\mathcal{P}$ observes that BL_j is the latest block on the blockchain and $Tx_{\mathbb{CM},i}$ has been published through the block BL_k with $k \leq j$ but its first output has not been spent by $Tx_{\mathbb{RV},i}$.
2. if $j + 1 - k < t$:
   - $\mathcal{P}$ publishes $Tx_{\mathbb{RV},i}$ on the blockchain.

   Otherwise
   - $\mathcal{P}$ assigns the l^{th} element of D from witness $\gamma^1_{\mathbb{CM},i}$ to $\sigma^{\mathcal{P}}_{\mathbb{CM},i}$ where $l := 1$ if $\mathcal{P} = A$ and $l := 2$ otherwise.
   - $\mathcal{P}$ computes $y_{\bar{\mathcal{P}},i} = \mathsf{Ext}(\sigma^{\mathcal{P}}_{\mathbb{CM},i}, \tilde{\sigma}^{\mathcal{P}}_{\mathbb{CM},i}, Y_{\bar{\mathcal{P}},i})$.
   - If $Tx_{\mathbb{RC}}$ is unpublished:
     - $\mathcal{P}$ computes $\sigma^{\mathcal{P},1}_{\mathbb{PN}_1,i} = \mathsf{Sign}_{sk_{\mathcal{P}}}(f([Tx_{\mathbb{PN}_1,i}], 1))$ and $\sigma^{\bar{\mathcal{P}}_Y,1}_{\mathbb{PN}_1,i} = \mathsf{Sign}_{y_{\bar{\mathcal{P}},i}}(f([Tx_{\mathbb{PN}_1,i}], 1))$.
     - $\mathcal{P}$ creates $Tx_{\mathbb{PN}_1,i}$ using $[Tx_{\mathbb{PN}_1,i}]$, $\gamma^1_{\mathbb{PN}_1,i} = (j, (\sigma^{\mathcal{P},1}_{\mathbb{PN}_1,i}, \sigma^{\bar{\mathcal{P}}_Y,1}_{\mathbb{PN}_1,i}))$ with $j = 1$ if $\mathcal{P} = B$ or $j = 3$ otherwise and $\gamma^2_{\mathbb{PN}_1,i} = (1, (\sigma^{\mathcal{A},2}_{\mathbb{PN}_1,i}, \sigma^{\mathcal{B},2}_{\mathbb{PN}_1,i}, \sigma^{\mathcal{W},2}_{\mathbb{PN}_1,i}))$.
     - $\mathcal{P}$ publishes $Tx_{\mathbb{PN}_1,i}$ on-chain.
   - Else:
     - $\mathcal{P}$ computes $\sigma^{\mathcal{P},1}_{\mathbb{PN}_2,i} = \mathsf{Sign}_{sk_{\mathcal{P}}}(f([Tx_{\mathbb{PN}_2,i}], 1))$ and $\sigma^{\bar{\mathcal{P}}_Y,1}_{\mathbb{PN}_2,i} = \mathsf{Sign}_{y_{\bar{\mathcal{P}},i}}(f([Tx_{\mathbb{PN}_2,i}], 1))$.

- $\mathcal{P}$ creates $Tx_{\mathbb{PN}_2,i}$ using $[Tx_{\mathbb{PN}_2,i}]$, $\gamma^1_{\mathbb{PN}_2,i} = (j, (\sigma^{\mathcal{P},1}_{\mathbb{PN}_2,i}, \sigma^{\bar{\mathcal{P}}_Y,1}_{\mathbb{PN}_2,i}))$ with $j = 1$ if $\mathcal{P} = B$ or $j = 3$ otherwise and $\gamma^2_{\mathbb{PN}_2,i} = (1, (\sigma^{\mathcal{A},2}_{\mathbb{PN}_2,i}, \sigma^{\mathcal{B},2}_{\mathbb{PN}_2,i}, \sigma^{\mathcal{W},2}_{\mathbb{PN}_2,i}))$.
- $\mathcal{P}$ publishes $Tx_{\mathbb{PN}_2,i}$ on-chain.

D Temporarily unavailable watchtower

FPPW can adapt to situations where channel parties want to update the channel state but watchtower is temporarily unavailable. A way to deal with such occasions is that channel parties wait for the watchtower to get responsive. However, it disturbs the main functionality of the payment channel. Another solution is updating the channel with skipping those steps that require the watchtower to sign the new revocation transaction (see step 3 of the channel update protocol in Appendix C) and new penalty transactions (see step 7 of the channel update protocol in Appendix C). This solution also has two problems. Firstly, without the watchtower cooperation, the new commit transaction gets irrevocable. Thus, it is impossible for the channel to be updated several times. Secondly, if the watchtower gets uncooperative, channel parties will be forced to update the channel on-chain even if they both agree to remain always online and continue using the channel.

To resolve this issue, channel parties can take steps of the channel update phase using the following commit transactions:

$$Tx_{\mathbb{CM},i+1} = [(O_{\mathbb{FU}} \| 1)] \to [(a + b \mid \varphi^1_{\mathbb{CM},i+1}), (\epsilon \mid \varphi^2_{\mathbb{CM},i+1})] \tag{10}$$

where

$$\varphi^1_{\mathbb{CM},i+1} = \varphi^1_1 \vee \varphi^1_2 \vee \varphi^1_3$$

with $\varphi^1_1 = pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge \Delta_{2t}$, $\varphi^1_2 = pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge pk_{\mathcal{W}}$ and $\varphi^1_3 = pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge \Delta_t$ and

$$\varphi^2_{\mathbb{CM},i+1} = \varphi^2_1 \vee \varphi^2_2 \vee \varphi^2_3$$

with $\varphi^2_1 = pk_{\mathcal{B}} \wedge Y_{\mathcal{A},i+1} \wedge \Delta_t$, $\varphi^2_2 = pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge pk_{\mathcal{W}}$ and $\varphi^2_3 = pk_{\mathcal{A}} \wedge Y_{\mathcal{B},i+1} \wedge \Delta_t$

Also, a new type of revocation transaction is introduced as follows:

$$Tx_{\mathbb{RV}',i+1} = [(O^1_{\mathbb{CM},i+1} \| 3)] \to [(a + b \mid Y_{\mathcal{A},i+1} \wedge Y_{\mathcal{B},i+1})], \tag{11}$$

Then, the update protocol is as follows:

Preconditions: The channel establishment phase is complete and $Tx_{\mathbb{FU}}$ and $Tx_{\mathbb{CL}}$ are on-chain. The channel update phase has completed i times and hence the channel is at state i. The watchtower is unavailable.

1. Create $[Tx_{\mathbb{CM},i+1}]$:

   (a) Party $\mathcal{P}$ generates $(Y_{\mathcal{P},i+1}, y_{\mathcal{P},i+1}) \leftarrow \mathsf{GenR}$.
   (b) $\mathcal{P} \xhookrightarrow{Y_{\mathcal{P},i+1}} \bar{\mathcal{P}}$
   (c) If Party $\mathcal{P}$ receives $Y_{\bar{\mathcal{P}},i+1}$, it creates $[Tx_{\mathbb{CM},i+1}]$ according to 10. Else it stops.
2. Create $[Tx_{\mathbb{SP},i+1}]$: Party $\mathcal{P}$ creates $[Tx_{\mathbb{SP},i+1}]$ according to 3.
3. Create $[Tx_{\mathbb{RV}',i+1}]$: Party $\mathcal{P}$ creates $[Tx_{\mathbb{RV}',i+1}]$ according to 11.
4. Create $Tx_{\mathbb{SP},i+1}$:
   (a) Party $\mathcal{P}$ computes $\sigma^{\mathcal{P}}_{\mathbb{SP},i+1} = \mathsf{Sign}_{sk_{\mathcal{P}}}([f(Tx_{\mathbb{SP},i+1}], 1))$.
   (b) $\mathcal{P} \xhookrightarrow{\sigma^{\mathcal{P}}_{\mathbb{SP},i+1}} \bar{\mathcal{P}}$
   (c) If party $\mathcal{P}$ receives $\sigma^{\bar{\mathcal{P}}}_{\mathbb{SP},i+1}$ s.t. $\mathsf{Vrfy}_{pk_{\bar{\mathcal{P}}}}(f([Tx_{\mathbb{SP},i+1}], 1); \sigma^{\bar{\mathcal{P}}}_{\mathbb{SP},i+1}) = 1$, it continues. Else it stops.
   (d) Party $\mathcal{P}$ creates $Tx_{\mathbb{SP},i+1}$.
5. Create $Tx_{\mathbb{CM},i+1}$:
   (a) Party $\mathcal{P}$ computes $\tilde{\sigma}^{\mathcal{P}}_{\mathbb{CM},i+1} = \mathsf{pSign}_{sk_{\mathcal{P}}}(f([Tx_{\mathbb{CM},i+1}], 1), Y_{\bar{\mathcal{P}},i+1})$.
   (b) $\mathcal{P} \xhookrightarrow{\tilde{\sigma}^{\mathcal{P}}_{\mathbb{CM},i+1}} \bar{\mathcal{P}}$
   (c) If party $\mathcal{P}$ receives $\tilde{\sigma}^{\bar{\mathcal{P}}}_{\mathbb{CM},i+1}$ s.t. $\mathsf{pVrfy}_{pk_{\bar{\mathcal{P}}}}(f([Tx_{\mathbb{CM},i+1}], 1), Y_{\mathcal{P},i+1}; \tilde{\sigma}^{\bar{\mathcal{P}}}_{\mathbb{CM},i+1}) = 1$, it computes $\sigma^{\bar{\mathcal{P}}}_{\mathbb{CM},i+1} = \mathsf{Adapt}(\tilde{\sigma}^{\bar{\mathcal{P}}}_{\mathbb{CM},i+1}, y_{\mathcal{P},i+1})$, computes $\sigma^{\mathcal{P}}_{\mathbb{CM},i+1} = \mathsf{Sign}_{sk_{\mathcal{P}}}$ $(f([Tx_{\mathbb{CM},i+1}], 1))$, creates $Tx_{\mathbb{CM},i+1}$ and continues. Else it execute the non-collaborative closure phase (from $\mathcal{P}$'s point of view the channel is still at state i).
6. Create $Tx_{\mathbb{RV},i}$ or $Tx_{\mathbb{RV}',i}$:
   (a) If $Tx_{\mathbb{CM},i}$ is according to 2,
      i. Party $\mathcal{P}$ computes $\sigma^{\mathcal{P},1}_{\mathbb{RV},i} = \mathsf{Sign}_{sk_{\mathcal{P}}}(f([Tx_{\mathbb{RV},i}], 1))$ as part of $\gamma^{\mathcal{P},1}_{\mathbb{RV},i}$ and $\sigma^{\mathcal{P},2}_{\mathbb{RV},i} = \mathsf{Sign}_{sk_{\mathcal{P}}}(f([Tx_{\mathbb{RV},i}], 2))$ as part of $\gamma^{2}_{\mathbb{RV},i}$.
      ii. $\mathcal{P} \xhookrightarrow{\sigma^{\mathcal{P},1}_{\mathbb{RV},i}, \sigma^{\mathcal{P},2}_{\mathbb{RV},i}} \bar{\mathcal{P}}$
      iii. If party $\mathcal{P}$ receives $\sigma^{\bar{\mathcal{P}},1}_{\mathbb{RV},i}$ and $\sigma^{\bar{\mathcal{P}},2}_{\mathbb{RV},i}$ from $\bar{\mathcal{P}}$ s.t. $\mathsf{Vrfy}_{pk_{\bar{\mathcal{P}}}}(f([Tx_{\mathbb{RV},i}], 1); \sigma^{\bar{\mathcal{P}},1}_{\mathbb{RV},i}) = 1$ and $\mathsf{Vrfy}_{pk_{\bar{\mathcal{P}}}}(f([Tx_{\mathbb{RV},i}], 2); \sigma^{\bar{\mathcal{P}},2}_{\mathbb{RV},i}) = 1$, it continues. Else it executes the non-collaborative closure phase (The channel now is at state $i + 1$).
      iv. Party $\mathcal{P}$ creates $Tx_{\mathbb{RV},i}$.
   (b) Otherwise:
      i. Party $\mathcal{P}$ computes $\sigma^{\mathcal{P},1}_{\mathbb{RV}',i} = \mathsf{Sign}_{sk_{\mathcal{P}}}(f([Tx_{\mathbb{RV}',i}), 1])$ as part of $\gamma^{\mathcal{P},1}_{\mathbb{RV}',i}$.
      ii. $\mathcal{P} \xhookrightarrow{\sigma^{\mathcal{P},1}_{\mathbb{RV}',i}} \bar{\mathcal{P}}$.

iii. If party $\mathcal{P}$ receives $\sigma^{\bar{\mathcal{P}},1}_{\mathbb{RV}',i}$ from $\bar{\mathcal{P}}$ s.t. $\mathsf{Vrfy}_{pk_{\bar{\mathcal{P}}}}(f([Tx_{\mathbb{RV}',i}], 1); \sigma^{\bar{\mathcal{P}},1}_{\mathbb{RV}',i}) = 1$, it continues. Else it executes the non-collaborative closure phase (The channel now is at state $i+1$).
iv. Party $\mathcal{P}$ creates $Tx_{\mathbb{RV}',i}$.

Remark 6. The first (second) condition in step 6 corresponds with the case when watchtower was available (unavailable) during the latest channel update.

Now assume that one of the revoked $Tx_{\mathbb{CM},i}$, for which $Tx_{\mathbb{RV}',i}$ has been created, is published by a channel party. The cheating party must wait $2t$ rounds to be able to publish the split transaction $Tx_{\mathbb{SP},i}$. However, its online counterparty can wait for t rounds and then publish the $Tx_{\mathbb{RV}',i}$ and take all the funds of the channel. Steps of this procedure are as following:

Preconditions: The channel establishment phase is complete and $Tx_{\mathbb{FU}}$ and $Tx_{\mathbb{CL}}$ are on-chain. The channel update phase has successfully completed n times. $Tx_{\mathbb{CM},i}$ with $i < n$ is recorded on-chain. Parties have created $Tx_{\mathbb{RV}',i}$. Channel parties are always online.

1. $\mathcal{P}$ observes that $Tx_{\mathbb{CM},i}$ is on-chain. $\mathcal{P}$ waits for t blocks.
2. $\mathcal{P}$ publishes $Tx_{\mathbb{RV}',i}$ on-chain.

When the watchtower becomes available, $Tx_{\mathbb{RV},i}$, $Tx_{\mathbb{PN}_1,i}$ and $Tx_{\mathbb{PN}_2,i}$ (respectively according to 6, 7 and 8) for the new agreed states can be created by $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{W}$. Steps of this procedure are as following:

Preconditions: The channel establishment phase is complete and $Tx_{\mathbb{FU}}$ and $Tx_{\mathbb{CL}}$ are on-chain. The channel update phase has successfully completed m times with $m \geq 1$. The watchtower $\mathcal{W}$ was unavailable during the latest k channel updates with $0 < k \leq m$. The watchtower $\mathcal{W}$ is now available.

1. $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{W}$ repeats the following steps for $i = m-k+1$ to $i = m-1$:
   (a) Create $[Tx_{\mathbb{RV},i}]$:
      i. $\mathcal{W}$ creates $[Tx_{\mathbb{RV},i}]$ according to 6.
      ii. $\mathcal{W}$ computes $\sigma^{\mathcal{W},j}_{\mathbb{RV},i} = \mathsf{Sign}_{sk_{\mathcal{W}}}([f(Tx_{\mathbb{RV},i}], j))$ with $j = \{1, 2\}$.
      iii. $\mathcal{W} \xhookrightarrow{\sigma^{\mathcal{W},1}_{\mathbb{RV},i}, \sigma^{\mathcal{W},2}_{\mathbb{RV},i}} \mathcal{A}, \mathcal{B}$.
      iv. If party $\mathcal{P} \in \{\mathcal{A}, \mathcal{B}\}$ receives $\sigma^{\mathcal{W},1}_{\mathbb{RV},i}$ and $\sigma^{\mathcal{W},2}_{\mathbb{RV},i}$ from $\mathcal{W}$ s.t. $\mathsf{Vrfy}_{pk_{\mathcal{W}}}(f([Tx_{\mathbb{RV},i}], j); \sigma^{\mathcal{W},j}_{\mathbb{RV},i}) = 1$ with $j = \{1, 2\}$, it continues. Else it stops.
   (b) Create $Tx_{\mathbb{RV},i}$:
      i. Party $\mathcal{P} \in \{\mathcal{A}, \mathcal{B}\}$ creates $[Tx_{\mathbb{RV},i}]$ according to 6.

ii. Party $\mathcal{P}$ computes $\sigma^{\mathcal{P},1}_{\mathbb{RV},i} = \mathsf{Sign}_{sk_{\mathcal{P}}}(f([Tx_{\mathbb{RV},i}],1))$ as part of $\gamma^1_{\mathbb{RV},i}$ and $\sigma^{\mathcal{P},2}_{\mathbb{RV},i} = \mathsf{Sign}_{sk_{\mathcal{P}}}(f([Tx_{\mathbb{RV},i}],2))$ as part of $\gamma^2_{\mathbb{RV},i}$.

iii. $\mathcal{P} \xhookrightarrow{\sigma^{\mathcal{P},1}_{\mathbb{RV},i},\sigma^{\mathcal{P},2}_{\mathbb{RV},i}} \bar{\mathcal{P}}$

iv. If party $\mathcal{P}$ receives $\sigma^{\bar{\mathcal{P}},1}_{\mathbb{RV},i}$ and $\sigma^{\bar{\mathcal{P}},2}_{\mathbb{RV},i}$ from $\bar{\mathcal{P}}$ s.t. $\mathsf{Vrfy}_{pk_{\bar{\mathcal{P}}}}(f([Tx_{\mathbb{RV},i}],1);\sigma^{\bar{\mathcal{P}},1}_{\mathbb{RV},i}) = 1$ and $\mathsf{Vrfy}_{pk_{\bar{\mathcal{P}}}}(f([Tx_{\mathbb{RV},i}],2);\sigma^{\bar{\mathcal{P}},2}_{\mathbb{RV},i}) = 1$, it continues. Else it executes the non-collaborative closure phase (The channel now is at state m).

v. $\mathcal{P} \xhookrightarrow{\sigma^{\mathcal{P},1}_{\mathbb{RV},i},\sigma^{\mathcal{P},2}_{\mathbb{RV},i}} \mathcal{W}$.

vi. If $\mathcal{W}$ receives $\sigma^{\mathcal{P},1}_{\mathbb{RV},i}$ and $\sigma^{\mathcal{P},2}_{\mathbb{RV},i}$ from $\mathcal{P} = \{\mathcal{A},\mathcal{B}\}$ s.t. $\mathsf{Vrfy}_{pk_{\mathcal{P}}}(f([Tx_{\mathbb{RV},i}],1);\sigma^{\mathcal{P},1}_{\mathbb{RV},i}) = 1$ and $\mathsf{Vrfy}_{pk_{\mathcal{P}}}(f([Tx_{\mathbb{RV},i}],2);\sigma^{\mathcal{P},2}_{\mathbb{RV},i}) = 1$, it continues. Else it stops.

vii. $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{W}$ create $Tx_{\mathbb{RV},i}$ using $[Tx_{\mathbb{RV},i}]$ and $\gamma^j_{\mathbb{RV},i} = (2,(\sigma^{\mathcal{A},j}_{\mathbb{RV},i},\sigma^{\mathcal{B},j}_{\mathbb{RV},i},\sigma^{\mathcal{W},j}_{\mathbb{RV},i}))$ with $j=\{1,2\}$.

(c) Create $[Tx_{\mathbb{PN}_1,i}]$ and $[Tx_{\mathbb{PN}_2,i}]$

i. Party $\mathcal{P} \in \{\mathcal{A},\mathcal{B}\}$ creates $[Tx_{\mathbb{PN}_1,i}]$ and $[Tx_{\mathbb{PN}_2,i}]$ according to 7 and 8.

ii. Party $\mathcal{P}$ computes $\sigma^{\mathcal{P},2}_{\mathbb{PN}_1,i} = \mathsf{Sign}_{sk_{\mathcal{P}}}(f([Tx_{\mathbb{PN}_1,i}],2))$ and $\sigma^{\mathcal{P},2}_{\mathbb{PN}_2,i} = \mathsf{Sign}_{sk_{\mathcal{P}}}(f([Tx_{\mathbb{PN}_2,i}],2))$ as part of $\gamma^2_{\mathbb{PN}_1,i}$ and $\gamma^2_{\mathbb{PN}_2,i}$ respectively.

iii. $\mathcal{P} \xhookrightarrow{\sigma^{\mathcal{P},2}_{\mathbb{PN}_1,i},\sigma^{\mathcal{P},2}_{\mathbb{PN}_2,i}} \bar{\mathcal{P}}$

iv. If $\mathcal{P}$ receives signatures $\sigma^{\bar{\mathcal{P}},2}_{\mathbb{PN}_1,i}$ and $\sigma^{\bar{\mathcal{P}},2}_{\mathbb{PN}_2,i}$ s.t. $\mathsf{Vrfy}_{pk_{\bar{\mathcal{P}}}}(f([Tx_{\mathbb{PN}_1,i}],2);\sigma^{\bar{\mathcal{P}},2}_{\mathbb{PN}_1,i}) = 1$ and $\mathsf{Vrfy}_{pk_{\bar{\mathcal{P}}}}(f([Tx_{\mathbb{PN}_2,i}],2);\sigma^{\bar{\mathcal{P}},2}_{\mathbb{PN}_2,i}) = 1$, it continues. Else it executes the non-collaborative closure phase (The channel now is at state m) or gets online at least once every $t-1$ blocks.

v. $\mathcal{W}$ creates $[Tx_{\mathbb{PN}_1,i}]$ and $[Tx_{\mathbb{PN}_2,i}]$ according to 7 and 8

vi. $\mathcal{W}$ computes $\sigma^{\mathcal{W},2}_{\mathbb{PN}_1,i} = \mathsf{Sign}_{sk_{\mathcal{W}}}(f([Tx_{\mathbb{PN}_1,i}],2)$ and $\sigma^{\mathcal{W},2}_{\mathbb{PN}_2,i} = \mathsf{Sign}_{sk_{\mathcal{W}}}(f([Tx_{\mathbb{PN}_2,i}],2)$.

vii. $W \xhookrightarrow{\sigma^{\mathcal{W},2}_{\mathbb{PN}_1,i},\sigma^{\mathcal{W},2}_{\mathbb{PN}_2,i}} \mathcal{P}$ with $\mathcal{P} = \{\mathcal{A},\mathcal{B}\}$.

viii. If $\mathcal{P} \in \{\mathcal{A},\mathcal{B}\}$ receives signatures $\sigma^{\mathcal{W},2}_{\mathbb{PN}_1,i}$ and $\sigma^{\mathcal{W},2}_{\mathbb{PN}_2,i}$ from $\mathcal{W}$ s.t. $\mathsf{Vrfy}_{pk_{\mathcal{W}}}(f([Tx_{\mathbb{PN}_1,i}],2);\sigma^{\mathcal{W},2}_{\mathbb{PN}_1,i}) = 1$ and $\mathsf{Vrfy}_{pk_{\mathcal{W}}}(f([Tx_{\mathbb{PN}_2,i}],2);\sigma^{\mathcal{W},2}_{\mathbb{PN}_2,i}) = 1$ it continues. Else it executes the non-collaborative closure phase (The channel now is at state m) or gets online at least once every $t-1$ blocks.

2. Create $[Tx_{\mathbb{RV},m}]$:

(a) $\mathcal{W}$ creates $[Tx_{\mathbb{RV},m}]$ according to 6.
(b) $\mathcal{W}$ computes $\sigma^{\mathcal{W},j}_{\mathbb{RV},m} = \mathsf{Sign}_{sk_{\mathcal{W}}}([f(Tx_{\mathbb{RV},m}], j))$ with $j = \{1, 2\}$.
(c) $\mathcal{W} \xhookrightarrow{\sigma^{\mathcal{W},1}_{\mathbb{RV},m}, \sigma^{\mathcal{W},2}_{\mathbb{RV},m}} \mathcal{A}, \mathcal{B}$.
(d) If party $\mathcal{P} \in \{\mathcal{A}, \mathcal{B}\}$ receives $\sigma^{\mathcal{W},1}_{\mathbb{RV},m}$ and $\sigma^{\mathcal{W},2}_{\mathbb{RV},m}$ from $\mathcal{W}$ s.t. $\mathsf{Vrfy}_{pk_{\mathcal{W}}}(f([Tx_{\mathbb{RV},m}], j); \sigma^{\mathcal{W},j}_{\mathbb{RV},m}) = 1$ with $j = \{1, 2\}$, it continues. Else it stops.

The commit and revocation transactions introduced in 10 and 11 have an interesting property that allows the honest channel party to penalize both the cheating party and the unresponsive watchtower. Assume that there is a revoked commit transaction $Tx_{\mathbb{CM},i}$ according to 10 for which the watchtower has also signed penalty transactions. If this transaction is published by a cheating party, let's say $\mathcal{A}$, the watchtower can immediately publish $Tx_{\mathbb{RV},i}$ to invalidate the penalty transactions. However, if the watchtower is unresponsive, after t rounds, $\mathcal{B}$ can firstly publish $Tx_{\mathbb{RV}',i}$ to penalize the cheating party $\mathcal{B}$ and then publish $Tx_{\mathbb{PN}_1,i}$ or $Tx_{\mathbb{PN}_2,i}$ to penalize the unresponsive watchtower. Split transaction $Tx_{\mathbb{SP},i}$ cannot be published within $2t - 1$ rounds of broadcast of $Tx_{\mathbb{CM},i}$.

E FPPW Transactions Scripts

Funding transaction has one output with the following script where pubkeyA, pubkeyB are public keys of $\mathcal{A}$ and $\mathcal{B}$, respectively:

```
2 ⟨pubkeyA⟩ ⟨pubkeyB⟩ 2 OP_CHECKMULTISIG
```

Commit transaction has one input that takes the output of the funding transaction with witness script 0 ⟨pubkeyA_sig⟩ ⟨pubkeyB_sig⟩. It also has two outputs where the script of its first output (main output) is as following:

```
OP_IF
   # Revocation
   3 ⟨Rev_pubkeyA⟩ ⟨Rev_pubkeyB⟩ ⟨Rev_pubkeyW⟩ 3 OP_CHECKMULTISIG
OP_ELSE
   # Split
   ⟨delay t⟩ OP_CHECKSEQUENCEVERIFY OP_DROP
   2 ⟨Spl_pubkeyA⟩ ⟨Spl_pubkeyB⟩ 2
OP_ENDIF
```

where ⟨Rev_pubkeyA⟩ and ⟨Spl_pubkeyA⟩ are public keys of $\mathcal{A}$, ⟨Rev_pubkeyB⟩ and ⟨Spl_pubkeyB⟩ are public keys of $\mathcal{B}$ and ⟨Rev_pubkeyW⟩ is public key of $\mathcal{W}$.

The script for the second output (auxiliary output) of the commit transaction is as following:

```
OP_IF
   # Revocation
   3 ⟨Rev_pubkeyA⟩ ⟨Rev_pubkeyB⟩ ⟨Rev_pubkeyW⟩ 3 OP_CHECKMULTISIG
OP_ELSE
   ⟨delay t⟩ OP_CHECKSEQUENCEVERIFY OP_DROP
```

```
  OP_IF
    # Penalty1 or Penalty2 by party B
    2 ⟨Pen_pubkeyB⟩ ⟨YA⟩ 2 OP_CHECKMULTISIG
  OP_ELSE
    # Penalty1 or Penalty2 by party A
    2 ⟨Pen_pubkeyA⟩ ⟨YB⟩ 2 OP_CHECKMULTISIG
  OP_ENDIF
OP_ENDIF
```

where ⟨Rev_pubkeyA⟩ and ⟨Pen_pubkeyA⟩ are public keys of $\mathcal{A}$, ⟨Rev_pubkeyB⟩ and ⟨Pen_pubkeyB⟩ are public keys of $\mathcal{B}$ and ⟨Rev_pubkeyW⟩ is public key of $\mathcal{W}$. Also, YA and YB are statement of $\mathcal{A}$ and $\mathcal{B}$, respectively. The witness script for input of split transaction is 0 ⟨Spl_pubkeyA_Sig⟩ ⟨Spl_pubkeyB_Sig⟩ 0

The revocation transaction has two inputs where its first and second inputs take the first and second outputs of the corresponding commit transaction, respectively. The witness script for both inputs is 0 ⟨Rev_pubkeyA_sig⟩ ⟨Rev_pubkeyB_sig⟩ ⟨Rev_pubkeyW_sig⟩ 1. It also has one output with the following script:

```
2 ⟨YA⟩ ⟨YB⟩ 2 OP_CHECKMULTISIG
```

Collateral transaction has one output with script 3 ⟨pubkeyA⟩ ⟨pubkeyB⟩ ⟨pubkeyW⟩ 3 OP_CHECKMULTISIG, where pubkeyA, pubkeyB and pubkeyW are the public keys of $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{W}$, respectively.

The reclaim transaction has one input taking the collateral transaction output with witness script:

```
0 ⟨pubkeyA_sig⟩ ⟨pubkeyB_sig⟩ ⟨pubkeyW_sig⟩
```

It also has a single output, with the following script:

```
OP_IF
  # Penalty2
  3 ⟨Pen_pubkeyA⟩ ⟨Pen_pubkeyB⟩ ⟨Pen_pubkeyW⟩ 3 OP_CHECKMULTISIG
OP_ELSE
  # normal
  ⟨delay T⟩ OP_CHECKSEQUENCEVERIFY OP_DROP
  ⟨pubkeyW⟩ OP_CHECKSIG
OP_ENDIF
```

where Pen_pubkeyA, Pen_pubkeyB and Pen_pubkeyW are public keys of $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{W}$, respectively. pubkeyW is also the public key of $\mathcal{W}$.

The penalty transaction 1 has two inputs. The first one takes the second output of the corresponding commit transaction with the following witness script, if $\mathcal{B}$ is broadcasting the penalty transaction:

```
0 ⟨Pen_pubkeyB_Sig⟩ ⟨YA_Sig⟩ 1 0
```

or with the following witness script, if $\mathcal{A}$ is broadcasting it:

```
0 ⟨Pen_pubkeyA_Sig⟩ ⟨YB_Sig⟩ 0 0
```

The second input takes the output of the collateral transaction with the witness script 0 ⟨pubkeyA_Sig⟩ ⟨pubkeyB_Sig⟩ ⟨pubkeyW_Sig⟩. Also, the script for its output is similar to that of the revocation transaction.

The penalty transaction 2 is similar to penalty transaction 1. The only difference is that its second input spends the output of the reclaim transaction. The witness for the second input is 0 ⟨Pen_pubkeyA_Sig⟩ ⟨Pen_pubkeyB_Sig⟩ ⟨Pen_pubkeyW_Sig⟩

F FPPW with One Hiring Party

If only one of the channel parties is going to hire the watchtower, some changes must be applied to FPPW. The transaction flows for FPPW in such scenarios is depicted in Fig. 7. In this scenario, without loss of generality, we assume that party $\mathcal{A}$ is the hiring party and hence only $\mathcal{A}$ funds the extra ϵ. Also, auxiliary output of the commit transaction has only two subconditions where the first one is used by $\mathcal{A}$ for penalty purposes and the second one can be used by both parties for revocation purposes. Output condition for $Tx_{\mathbb{CL}}$ as well as the first subcondition for output of $Tx_{\mathbb{RC}}$ is $pk_{\mathcal{A}} \wedge pk_{\mathcal{W}}$ and public key of party $\mathcal{B}$ is not involved in these transactions anymore. Furthermore, output of $Tx_{\mathbb{PN}_1,i}$ and $Tx_{\mathbb{PN}_2,i}$ can be only claimed by party $\mathcal{A}$. Therefore, If party $\mathcal{B}$ publishes a revoked commit transaction, $\mathcal{W}$ can publish its corresponding revocation transaction and then only $\mathcal{A}$ can claim its output. Otherwise, $\mathcal{A}$ can penalize the watchtower by publishing either $Tx_{\mathbb{PN}_1,i}$ or $Tx_{\mathbb{PN}_2,i}$. Similarly, if $\mathcal{A}$ publishes a revoked commit transaction, $\mathcal{B}$ can broadcast its corresponding revocation transaction and claim all the channel funds. The watchtower is only paid by $\mathcal{A}$.

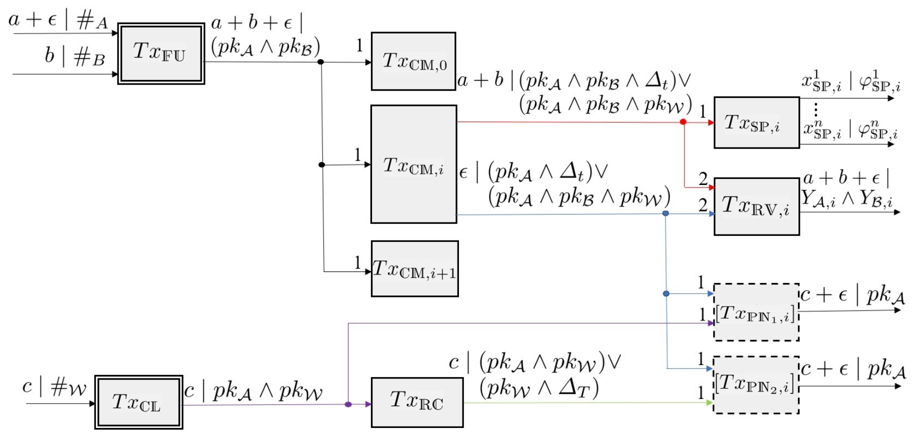

$a + \epsilon \mid \#_A$
$b \mid \#_B$
$Tx_{\mathbb{FU}}$
$a + b + \epsilon \mid$
$(pk_{\mathcal{A}} \wedge pk_{\mathcal{B}})$
1
$Tx_{\mathbb{CM},0}$
$a + b \mid (pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge \Delta_t) \vee$
$(pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge pk_{\mathcal{W}})$
1
$Tx_{\mathbb{CM},i}$
$\epsilon \mid (pk_{\mathcal{A}} \wedge \Delta_t) \vee$
$(pk_{\mathcal{A}} \wedge pk_{\mathcal{B}} \wedge pk_{\mathcal{W}})$
1
$Tx_{\mathbb{CM},i+1}$
1
$Tx_{\mathbb{SP},i}$
$x^1_{\mathbb{SP},i} \mid \varphi^1_{\mathbb{SP},i}$
$x^n_{\mathbb{SP},i} \mid \varphi^n_{\mathbb{SP},i}$
2
2
$Tx_{\mathbb{RV},i}$
$a + b + \epsilon \mid$
$Y_{\mathcal{A},i} \wedge Y_{\mathcal{B},i}$
1
1
$[Tx_{\mathbb{PN}_1,i}]$
$c + \epsilon \mid pk_{\mathcal{A}}$
1
1
$[Tx_{\mathbb{PN}_2,i}]$
$c + \epsilon \mid pk_{\mathcal{A}}$
$c \mid \#_{\mathcal{W}}$
$Tx_{\mathbb{CL}}$
$c \mid pk_{\mathcal{A}} \wedge pk_{\mathcal{W}}$
$Tx_{\mathbb{RC}}$
$c \mid (pk_{\mathcal{A}} \wedge pk_{\mathcal{W}}) \vee$
$(pk_{\mathcal{W}} \wedge \Delta_T)$

Fig. 7. An FPPW Bitcoin Channel with $\mathcal{A}$ being the Hiring Party.