Privacy-Preserving Cross-Chain Atomic Swaps

No Author Given

No Institute Given

Abstract. Recently, there has been a lot of interest in studying the transfer of assets across different blockchains in the form of *cross-chain atomic swaps*. Unfortunately, the current candidates of atomic swaps (hash-lock time contracts) offer no privacy; the identities as well as the exact trade that happened between any two parties is publicly visible.
In this work, we explore the different notions of privacy that we can hope for in an atomic swap protocol. Concretely, we define an atomic swap as a two-party protocol and formalize the different notions of privacy in the form of *anonymity*, *confidentiality* and *indistinguishability* of swap transactions.
As a building block, we abstract out the primitive of *Atomic Release of Secrets* (ARS) which captures atomic exchange of a secret for a predecided transaction. We then show how ARS can be used to build privacy-preserving cross-chain swaps.
We also show that the recently introduced notion of *adapter signatures* [Poe18, War17] is a concrete instantiation of ARS under the framework of Schnorr signatures [Sch91] and thus, construct a private cross-chain swap using Schnorr signatures.

1 Introduction

A key attraction of distributed ledgers is that they can replace a trusted party or an escrow service, for parties wishing to transact. Assets can be held and transferred according to programmable logic that the network evaluates a.k.a through a *smart contract*. A natural scenario to consider is the transfer of assets across different blockchains. Such a transfer is often referred to as an *cross-chain atomic swap*. Unfortunately, such a protocol offers no privacy; the identities as well as the exact trade that happened between any two parties is publicly visible. In this work, we explore the different notions of privacy that we can hope for in an atomic swap protocol. We might want to hide the identities of the parties involved in the swap so that we have an *anonymous* swap, we may want to hide the amounts transferred as part of the swap so that we have *confidential transactions* in a swap. We may also want to hide the fact that an atomic swap ever happened. These different notions of privacy may not be comparable and maybe specific to individual chains that are part of the swap, and in fact the different notions may have some trade-offs as we soon elaborate.

Suppose that Alice and Bob want to engage in an atomic cross-chain swap as follows: Alice is willing to trade her 5 ether for Bob's 2 BTC. Typically, they can proceed to do such an atomic swap through a hash-lock time contract

(HTLC) [PD16]. An HTLC is a contract with a hash value y associated with it, and on input a value s such that $H(s) = y$, the contract is executed and the specified amount gets transferred as per the contract. HTLC also has a time-out T associated such that if a valid s is not produced until time T then the amount is refunded to the party that initiates the contract.

The atomic swap protocol comprises of the following steps: First, Alice chooses a secret s and publishes on the Bitcoin blockchain an HTLC with $y = H(s)$ paying Bob 2 BTC, and with time-out T_B. After confirming that this contract satisfies pre-decided conditions, Bob publishes on the Ethereum blockchain an HTLC with the same y paying Alice 5 ether with timelock $T_A = T_B - \Delta$. Bob chooses Δ such that it leaves enough time for Bob to claim 2 BTC after Alice has claimed her ether amount. If the contract looks correct, Alice claims 5 ether and in the process reveals s. Bob can then claim the 2 BTC using the same s.

Unfortunately, such a protocol offers no privacy; the identities as well as the exact trade that happened between any two parties is publicly visible. In this work, we explore the different notions of privacy that we can hope for in an atomic swap protocol. We might want to hide the identities of the parties involved in the swap so that we have an *anonymous* swap, we may want to hide the amounts transferred as part of the swap so that we have *confidential transactions* in a swap. We may also want to hide the fact that an atomic swap ever happened. These different notions of privacy may not be comparable and maybe specific to individual chains that are part of the swap, and in fact the different notions may have some trade-offs as we soon elaborate.

1.1 Our Contributions

In this work, we initiate the study of privacy in cross-chain swaps. Concretely, our contributions are as follows:

– Formalizing Privacy in Cross-Chain Atomic Swaps. We define an atomic swap as a two-party protocol and formalize the different notions of privacy in the form of *anonymity*, *confidentiality* and *indistinguishability* of swap transactions.
– Private Swaps from Atomic Release of Secrets. We abstract out the primitive of *Atomic Release of Secrets* (ARS) which captures atomic exchange of a secret for a pre-decided transaction. We then show how ARS can be used to build privacy-preserving cross-chain swaps.
– Instantiating Atomic Release of Secrets. We show that the recently introduced notion of *adapter signatures* [Poe18, War17] is a concrete instantiation of ARS under the framework of Schnorr signatures [Sch91]. This in turn enables a private cross-chain swap using Schnorr signatures.

1.2 Related Work

Poon and Dryja [PD16] gave the first atomic swap protocol in the form of an HTLC contracts, several works [Nol13, BCD^{+}14] have extended that protocol.

Herlihy [Her18] formalized the notion of cross-chain atomic swaps for the first time, and generalized the definition to a swap across multiple parties. Cross-chain swap for some specific cases has been studied, for example Bitcoin and Ethereum [Pre18].

There has been a lot of work around ensuring fairness in atomic swaps [DEF, HLY19]. These works focus on appropriate punishments to the misbehaving party, whereas the guarantees that we want in a two-party swap are that either both assets are transferred or none of them are.

There has been extensive work on how different chains can communicate with each other [Kwo15, HHK18, Tho15]. Though privacy has been extensively studied in context of individual blockchains [SCG$^+$14, NM$^+$16, GM17, Max15], there has been no formal study of privacy in the context of cross-chain swaps.

1.3 A Simple Privacy-Preserving Atomic Cross-Chain Swap

As a starting point, we describe a protocol for an atomic cross-chain swap with the following privacy guarantee: It is impossible to link transactions across two chains that are part of an atomic swap. Moreover, the transactions on both the chains are indistinguishable from regular transactions on that chain.

Note that in the HTLC-based contract described above, the same hash value y is associated with both the contracts and hence, the transactions on both the chains are easily identifiable as being part of one atomic swap. We use a similar framework as above for our protocol, but instead of using the same value y, we use two different values y_A, y_B such that $z = y_A - y_B$ is computable only by Alice, Bob, and for any other observer the values y_A, y_B are unlinkable.

We assume public-key infrastructure (PKI); When Alice and Bob decide (off-chain) to execute an atomic swap, they can find each other's public key $\mathsf{pk}_A, \mathsf{pk}_B$. Alice and Bob can use the PKI to then execute a key-exchange protocol such as Diffie-Hellman key-exchange [DH76], to agree on a shared key.

We also assume that Alice and Bob agree on a (possibly pseudorandom) value $z \in \mathbb{Z}_p^*$ which could be a time-stamp of their off-chain communication or some function of key-exchange value of their public-keys (For instance, z can be a function of g^{ab} which is the value of Diffie-Hellman key-exchange between Alice and Bob). Note that this z is computable or known to only Alice and Bob.

The protocol is as follows:

1. Alice chooses a secret s_A and publishes on the Bitcoin blockchain an HTLC with $Y_A = g^{s_A}$ paying Bob 2 BTC, and with time-out T_B. Note that the hash function we are using here is modular exponentiation $H(s_A) = g^{s_A}$.
2. After confirming that this contract satisfies predecided conditions, Bob publishes on the Ethereum blockchain an HTLC with $y_B = y_A \cdot g^z = g^{s_A + z}$ paying Alice 5 ether with timelock $T_A = T_B - \Delta$.
3. If the contract looks correct, Alice computes $s_B = s_A + z$. She then claims 5 ether from Bob's contract, and in the process reveals s_B.
4. Bob can then claim the 2 BTC from Alice's contract using $s_A = s_B - z$.

Since z is computable or known only to Alice and Bob, the values s_A, s_B are unlinkable to any other observer, thus the transactions on either of the chains are indistinguishable from regular transactions.

2 Formalizing Privacy in Atomic Swaps

We will now formally define a private atomic swap protocol over two independent ledgers each with certain privacy properties. The underlying ledger will be captured through an abstract primitive of a *Transaction* over a distributed ledger.

Let coin be the native currency of the underlying ledger L. Let $\mathsf{tx}(A, B, x, L)$ denote a transaction that transfers x coins from A to B. A transaction tx should satisfy the following properties:

– *Correctness*: A tx is a proof of transfer of assets on the ledger, namely unconditional transfer of x coins from A to B on L.
– *Transaction Non-malleability.* This property requires that no bounded adversary can alter any of the data or any of the transactions published so far.
– *Balance.* This property requires that no bounded adversary can own more money than what he minted or received via payments from others.

A transaction may additionally satisfy following properties:

– *Confidentiality*: A transaction tx hides the amount of assets being transferred that is, two transactions between A and B for two different amounts look indistinguishable, denoted by $\approx$. More formally, for any two amounts x_0, x_1,
$$\mathsf{tx}(A, B, x_0, L) \approx \mathsf{tx}(A, B, x_1, L)$$
– *Anonymity*: A transaction tx hides the identities of the parties involved in it. More formally, for any two pairs of identities $(A_0, B_0), (A_1, B_1)$,
$$\mathsf{tx}(A_0, B_0, x, L) \approx \mathsf{tx}(A_1, B_1, x, L)$$

2.1 Private Atomic Swap Protocol

We will now formalize a private atomic swap protocol PAS over two ledgers L_1, L_2 with native currencies $\mathsf{coin}_1, \mathsf{coin}_2$ respectively. Assume that the swap takes place between parties Alice(A) and Bob(B), where Alice is willing to trade z_1 coin_1 with z_2 coin_2 of Bob. Such a protocol will be denoted by $\mathsf{PAS}(A, B, z_1, z_2, L_1, L_2)$. In more detail,

$$\mathsf{PAS}(A, B, z_1, z_2, L_1, L_2) \equiv \langle \mathsf{tx}_1(A, B, z_1, L_1), \mathsf{tx}_2(B, A, z_2, L_2)\rangle$$

where PAS can be characterized as a pair of transactions over L_1 and L_2 such that A transfers z_1 coin_1 to Bob ($\mathsf{tx}_1(A, B, z_1, L_1)$) and B transfers z_2 coin_2 to Alice ($\mathsf{tx}_2(B, A, z_2, L_2)$).

Informally the different properties that we want from an atomic swap protocol are as follows:

- *Correctness.* If both Alice and Bob follow the steps of the protocol correctly, then swap takes place with Alice receiving z_1 coin_1 and Bob receiving z_2 coin_2.
- *Soundness.* If either of the parties deviate from the protocol, the honest party does not end up worse off. More concretely, either both $\mathsf{tx}_1(A, B, z_1, L_1)$, $\mathsf{tx}_2(B, A, z_2, L_2)$ take place or neither.
- *Privacy (Indistinguishability of Swap Transactions).* The transactions that are part of the PAS protocol that is, $\mathsf{tx}_1, \mathsf{tx}_2$ should be indistinguishable to regular transactions on both the ledgers of L_1, L_2.
  We can decouple the privacy of the entire PAS protocol, and require *indistinguishability of swap transactions* to hold for either of the ledgers individually.
- *Confidential Swap.* Protocol PAS hides the amounts exchanged in the swap transaction.
- *Anonymous Swap.* Protocol PAS hides the identities of the parties involved in the swap transaction.

Note that the indistinguishability property for any ledger will also depend on the confidentiality and anonymity properties of that ledger. For example, all the amounts in a Bitcoin transaction are in clear, and Bitcoin does not offer any confidentiality. Hence, if an atomic swap protocol involves the Bitcoin chain and it satisfies indistinguishability of transactions with respect to Bitcoin, then the swap protocol cannot satisfy confidentiality of transactions.

Lemma 1. *Let* PAS *be a private atomic swap protocol over two ledgers* L_1, L_2*.* PAS *satisfies both* indistinguishability of transactions *and* confidentiality *if and only if the transactions in that ledger satisfy confidentiality.*

Lemma 2. *Let* PAS *be a private atomic swap protocol over two ledgers* L_1, L_2*.* PAS *satisfies both* indistinguishability of transactions *and* anonymity *if and only if the transactions in that ledger satisfy anonymity.*

Let us now formalize these privacy notions.

Privacy (Indistinguishability of Swap Transactions). Let $\mathsf{PAS}(A, B, z_1, z_2, L_1, L_2) \equiv \langle \mathsf{tx}_1(A, B, z_1, L_1), \mathsf{tx}_2(B, A, z_2, L_2)\rangle$ be the atomic swap protocol between parties Alice(A) and Bob(B), where Alice is willing to trade z_1 coin_1 with z_2 coin_2 of Bob. For both ledgers L_1, L_2 we require that for any A', B', z'_1, z'_2,

$$\mathsf{tx}(A, B, z_1, L_1) \approx \mathsf{tx}(A', B', z'_1, L_1) \text{ and } \mathsf{tx}(B, A, z_2, L_2) \approx \mathsf{tx}(B', A', z'_2, L_2)$$

Recall that $\mathsf{tx}(A', B', z'_1, L_1)$ and $\mathsf{tx}(B', A', z'_2, L_2)$ are transactions on L_1 and L_2 respectively.

Confidential Swap. Let $\langle \mathsf{tx}_1(A, B, z_1, L_1), \mathsf{tx}_2(B, A, z_2, L_2)\rangle$ be part of the $\mathsf{PAS}(A, B, z_1, z_2, L_1, L_2)$ protocol, and let $\langle \mathsf{tx}_1(A, B, z'_1, L_1), \mathsf{tx}_2(B, A, z'_2, L_2)\rangle$ be part of the $\mathsf{PAS}(A, B, z'_1, z'_2, L_1, L_2)$ protocol. For any A, B and for any amounts z_1, z_2, z'_1, z'_2, the following holds:

$$\langle \mathsf{tx}_1(A, B, z_1, L_1), \mathsf{tx}_2(B, A, z_2, L_2)\rangle \approx \langle \mathsf{tx}_1(A, B, z'_1, L_1), \mathsf{tx}_2(B, A, z'_2, L_2)\rangle$$

Anonymous Swap. Let $\langle \mathsf{tx}_1(A, B, z_1, L_1), \mathsf{tx}_2(B, A, z_2, L_2)\rangle$ be part of the $\mathsf{PAS}(A, B, z_1, z_2, L_1, L_2)$ protocol, and let $\langle \mathsf{tx}_1(A', B', z_1, L_1), \mathsf{tx}_2(B', A', z_2, L_2)\rangle$ be part of the $\mathsf{PAS}(A', B', z_1, z_2, L_1, L_2)$ protocol. For any z_1, z_2 and for any participants A, BA', B', the following holds:

$$\langle \mathsf{tx}_1(A, B, z_1, L_1), \mathsf{tx}_2(B, A, z_2, L_2)\rangle \approx \langle \mathsf{tx}_1(A', B', z_1, L_1), \mathsf{tx}_2(B', A', z_2, L_2)\rangle$$

3 Atomic Release of Secrets

We now define the new primitive of *Atomic Release of Secrets* (ARS) which is a two-party protocol that enables a *conditional exchange* between two entities without a trusted intermediary. The setting is as follows: Alice and Bob agree on a transaction tx that pays Alice some predecided amount z on a ledger L ($\mathsf{tx}(A, B, z, L)$). For example, suppose Bob agrees to pay Alice some Bitcoins for a rare audio recording that Alice has. The guarantee of an ARS protocol is that tx will be published on L if and only if Bob learns Alice's secret s. In the previous example, tx will be published on the Bitcoin blockchain if and only if Bob gets the audio file (or learns a link address that gives him access to the file). Such a primitive can be directly useful in realizing cross-chain atomic swaps as we elaborate later in the section.

3.1 Definition: Atomic Release of Secrets

Let Alice, Bob be two entities that agree on a transaction tx and let s be Alice's secret. Let com be a homomorphic commitment scheme. Let $\mathsf{ARS}(B, A, z, \mathsf{com}(s))$ denote an ARS protocol between Alice and Bob for $\mathsf{tx}(B, A, z, L)$ and secret s. Such a protocol is a valid ARS protocol if the following property holds:

Atomic Release: The transaction tx is published if and only if Bob learns Alice's secret s.

In other words, if Alice and Bob engage in an ARS protocol then it is not possible that tx is published and Bob does not learn s or vice versa. It also means that knowledge of secret s gives the power to publish the tx.

Privacy of ARS: The transaction tx is indistinguishable from a regular transaction on the ledger. In particular, there is no way to tell if any tx was part of an ARS protocol or not.

Theorem 1 (Informal). *Protocols* $\mathsf{ARS}(B, A, z, \mathsf{com}(s))$ *and* $\mathsf{PAS}(A, B, z_1, z_2, L_1, L_2)$ *are equivalent; One can implement either from the other.*

Proof Sketch.

Claim. $\mathsf{ARS}(A, B, z, \mathsf{com}(s))$ implies $\mathsf{PAS}(A, B, z_1, z_2, L_1, L_2)$

Let $\mathsf{PAS}(A, B, z_1, z_2, L_1, L_2) = \langle \mathsf{tx}_1(A, B, z_1), \mathsf{tx}_2(B, A, z_2)\rangle$ be the the private swap protocol that we want to execute. Bob chooses a secret s and initiates an ARS protocol as $\mathsf{ARS}(A, B, z_1, \mathsf{com}(s))$. Bob then sends $\mathsf{com}(s)$ to Alice through a private channel. Alice homomorphically computes $\mathsf{com}(s')$ for a secret invertible function $f(s) = s'$ of her choosing. She then initiates $\mathsf{ARS}(B, A, z_2, \mathsf{com}(s'))$. If Bob publishes tx_1 then by the property of the ARS, Alice learns s and correspondingly $s' = f(s)$ as well thereby publishing tx_2 atomically. Correctness, soundness and privacy of the PAS follow from the properties of the underlying ARS and the homomorphic commitments.

Claim. $\mathsf{PAS}(A, B, z_1, z_2, L_1, L_2)$ implies $\mathsf{ARS}(A, B, z, \mathsf{com}(s))$

Alice chooses secret s and sends $\mathsf{com}(s)$ to Bob. If Bob later learns secret s from Alice, he executes $\mathsf{PAS}(A, B, z_1, z_2, L_1, L_2)$.

3.2 Instantiation: Atomic Release of Secrets

We now describe an instantiation of ARS in the form of *Adapter Signatures*. These are based on the classic construction of Schnorr signatures.

Schnorr Signatures We first recall Schnorr Signatures [Sch91]. Let $\mathbb{G}$ be a group of prime order q with generator g and let $H : \{0,1\}^{\mathsf{poly}(k)} \to \{0,1\}^k$ be any collision-resistant hash function.

Key Generation $(s, g^s) \leftarrow \mathsf{KeyGen}(1^k)$ The key generation algorithm takes in the security parameter, and outputs secret key sk chosen uniformly as $s \leftarrow \mathbb{Z}_q^*$ and public key pk as g^s.

Signing $(R, \sigma) \leftarrow \mathsf{Com}(\mathsf{sk}, \mathsf{msg})$ The signing algorithm takes as input secret key sk and the message msg and outputs signature $\mathsf{sig} = (R, \sigma)$ computed as follows:

- Choose $r \leftarrow \mathbb{Z}_q^*$ and $R = g^r$.
- Compute $\sigma = r + H(pk \mid R \mid \mathsf{msg}) \cdot s$

Verification $0/1 \leftarrow \mathsf{Verify}(\mathsf{pk}, \mathsf{sig}, \mathsf{msg})$ The verification algorithm takes as input the public key pk, message msg and the signature sig and checks if

$$g^\sigma = R \cdot \mathsf{pk}^{H(pk \mid R \mid \mathsf{msg})}$$

Schnorr signatures are the primary tool for checking validity of transactions on Bitcoin as well as on most other blockchains. If Alice wants to transfer money to Bob, she needs to sign that transaction with her secret key. Anyone can then verify the signature confirming that Alice is the rightful owner of the account from which she is transferring money. If the signature verifies, then the transaction will be published on the blockchain.

3.3 Adapter Signatures

At a high level, an adaptor signature is a partial signature with a secret associated with it. It is a commitment such that if a predecided signature is published then the underlying secret of the adaptor signature can be derived. An adaptor signature functions as a kind of "promise" that a signature you agree to publish will reveal a secret. Such a signature can in turn be used for multiple applications such as atomic swaps as we elaborate later.

In more detail, an adapter signature is a protocol between two parties Alice and Bob that works as follows:

- Alice chooses $r, t \leftarrow \mathbb{Z}_q^*$ and sends $R = g^r, T = g^t$ to Bob.
- Bob computes the challenge $c = H(B \mid R + T \mid \mathsf{msg})$ and sends $c \cdot b$ to Alice, where (B, b) is the public-private key pair for Bob $(\mathsf{pk}_B, \mathsf{sk}_B)$ and msg is any message that Alice and Bob want to sign jointly.
- Alice can now compute the adapter signature $\sigma_{\mathsf{adapt}} = c \cdot b + r$ and send it to Bob.
- Note that σ_{adapt} is not a valid signature, but Bob can still verify the correctness of the adapter signature by checking:

$$g^{\sigma_{\mathsf{adapt}}} = R \cdot \mathsf{pk}_B^{H(pk_B \ \mid \ R+T \ \mid \ \mathsf{msg})}$$

  Alice now publishes a valid signature $\sigma = r + t + c \cdot b$ as part of a transaction published on the blockchain, and this signature can be verified by anyone. Once Bob sees this, Bob can derive the secret t as $\sigma - \sigma_{\mathsf{adapt}}$.

Theorem 2 (Informal). *The adapter signature protocol described above implies an* ARS *protocol for Alice and Bob.*

Proof Sketch. Alice and Bob agree on a desired transaction transferring assets from Bob to Alice and on secret t. This transaction needs a valid signature from Bob to be accepted by the blockchain. The adapter signature protocol guarantees that the signature is published if and only if Bob learns the secret t. Also the signature generated is a valid Schnorr signature, indistinguishable from any other signature, which ensures privacy of the corresponding ARS protocol.

3.4 Atomic Cross-Chain Swap using Adapter Signatures

We proved that adapter signatures directly give an ARS protocol for two parties Alice and Bob. We also proved that if there exists an ARS protocol and homomorphic commitments, then Alice and Bob can engage in a private cross-chain swap. Note that the signature itself acts as a homomorphic commitment in this case.

We now describe a concrete protocol for completeness. Recall that Alice is willing to trade z_1 coin_1 with z_2 coin_2 of Bob. The corresponding transactions are $\mathsf{tx}_1(A, B, z_1, L_1)$ and $\mathsf{tx}_2(B, A, z_2, L_2)$ respectively. The protocol will be as follows:

- Alice and Bob generate ephemeral (Schnorr) verification keys $\mathsf{pk}_A^1, \mathsf{pk}_A^2$ and $\mathsf{pk}_B^1, \mathsf{pk}_B^2$. Note that these are generated by choosing $s \leftarrow \mathbb{Z}_q^*$ and $\mathsf{pk} = g^s$. Thus, Alice knows a_1, a_2, whereas Bob knows b_1, b_2.
  The key for tx_1 is assigned as $\mathsf{pk}_A^1 + \mathsf{pk}_B^1$ and the key for tx_2 is assigned as $\mathsf{pk}_A^2 + \mathsf{pk}_B^2$. In other words, we set up two 2-out-of-2 multi-signature transactions.
- Alice chooses $t, r_1, r_2 \leftarrow \mathbb{Z}_q^*$. Let c_1, c_2 be the challenge for the two signatures corresponding to $\mathsf{tx}_1, \mathsf{tx}_2$ respectively. Alice sends $R_1 = g^{r_1}, R_2 = g^{r_2}, T = g^t$ and $c_1 \cdot a_1$ and $c_2 \cdot a_2$ to Bob.
- Bob adds his part of the keys to generate $c_1 \cdot (a_1 + b_1)$ and $c_2 \cdot (a_2 + b_2)$. Note that Bob can compute the challenges c_1, c_2 on his own.
- Alice creates two adapter signatures $\sigma^1_{\mathsf{adapt}} = r_1 + c_1 \cdot (a_1 + b_1)$ and $\sigma^2_{\mathsf{adapt}} = r_2 + c_2 \cdot (a_2 + b_2)$ and sends to Bob. Bob can verify both of these with respect to R_1, R_2 that Alice sent before.
- Finally, when Alice publishes tx_2 using $\sigma_2 = \sigma^2_{\mathsf{adapt}} + t$, that atomically reveals t to Bob and thus enabling him to publish $\sigma_1 = \sigma^1_{\mathsf{adapt}} + t$ and in turn, publishing tx_1.

4 Conclusion

In this work, we initiate the study of privacy in cross-chain atomic swaps. We formalize the different notions of privacy that we can expect from a cross-chain swap and we show the different trade-offs between these notions. In particular, we show how these notions of privacy depend heavily on the privacy of the underlying chains. We also give a concrete instantiation of a private swap protocol.

While we look at two-party swaps as a starting point, the next step is to study privacy of swaps across multiple chains. It may also be worthwhile to look at specific blockchains and study the exact privacy properties achievable for cross-chain swaps involving those chains.

References

[BCD$^+$14] Adam Back, Matt Corallo, Luke Dashjr, Mark Friedenbach, Gregory Maxwell, Andrew Miller, Andrew Poelstra, Jorge Timón, and Pieter Wuille. Enabling blockchain innovations with pegged sidechains. *URL: http://www. opensciencereview. com/papers/123/enablingblockchain-innovations-with-pegged-sidechains*, page 72, 2014.

[DEF] Stefan Dziembowski, Lisa Eckey, and Sebastian Faust. Fairswap: How to fairly exchange digital goods. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, pages 967–984.

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 1976.

[GM17] Matthew Green and Ian Miers. Bolt: Anonymous payment channels for decentralized currencies. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 473–489. ACM, 2017.

[Her18] Maurice Herlihy. Atomic cross-chain swaps. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*, pages 245–254. ACM, 2018.

[HHK18] Julian Hosp, Toby Hoenisch, and Paul Kittiwongsunthorn. COMIT - cryptographically-secure off-chain multi-asset instant transaction network. *CoRR*, 2018.

[HLY19] Runchao Han, Haoyu Lin, and Jiangshan Yu. On the optionality and fairness of atomic swaps. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, pages 62–75. ACM, 2019.

[Kwo15] Buchman E. Kwon, J. Cosmos: A network of distributed ledgers, 2015. `https://github.com/cosmos/cosmos/blob/master/WHITEPAPER.md`.

[Max15] G Maxwell. Confidential transactions, 2015. `https://people.xiph.org/~greg/confidential_values.txt`.

[NM+16] Shen Noether, Adam Mackenzie, et al. Ring confidential transactions. *Ledger*, 1:1–18, 2016.

[Nol13] T. Nolan. Alt chains and atomic transfers, 2013. `https://bitcointalk.org/index.php?topic=193281.msg2224949#msg2224949`.

[PD16] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments, 2016.

[Poe18] A. Poelstra. Mimblewimble and scriptless scripts, Real World Crypto, 2018. `https://www.youtube.com/watch?v=ovCBT1gyk9c&t=0s`.

[Pre18] James Prestwich. Non-atomic swaps, Bitcoin Expo, 2018. `https://www.youtube.com/watch?v=njGSFAOz7F8&feature=emb_logo`.

[SCG+14] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474. IEEE, 2014.

[Sch91] Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of cryptology*, 4(3):161–174, 1991.

[Tho15] Schwartz E Thomas, S. A protocol for interledger payments, 2015. `https://interledger.org/interledger.pdf`.

[War17] Warwing. Flipping the scriptless script on schnorr, 2017. `https://joinmarket.me/blog/blog/flipping-the-scriptless-script-on-schnorr/`.