TradeLayer: Peer-to-Peer Electronic Cash Backed By Decentralized Derivative Exchange

Patrick Dugan
desk@tradelayer.org
www.tradelayer.org

Abstract. Cryptocurrency trading has depended on custodial exchanges' corporate bank accounts, on-chain transaction volumes are dominated by batching between exchanges, fee-subsidized trading pollutes data-authenticity, and stable units of account suffer from reflexive instability or the regulatory capture of the banking system. Sidechains and the Lightning Network were proposed in 2015 as a way to desegregate transaction data-footprint from the Bitcoin blockchain's capacity. The OmniLayer protocol was developed from the first ICO's proceeds in 2013, attempting to realize a vision of decentralized exchange on the Bitcoin blockchain, playing host to Tether USD which has reached over two billion dollars in float by 2018, yet has not completed an audit. The TradeLayer protocol completes the vision for the Bitcoin family of protocols by extending OmniLayer to interact with Tendermint, providing Sidechain-like functionality for trade orders and payments. The protocol supports decentralized derivatives that clear in chains of bilateral participants, providing a secure, hedged basis for the issuance of audit-able, yield-bearing, fiat-denominated bearer currency redeemable in cryptocurrency. The clearing algorithm may be applicable to parallel implementations in the Lightning Network context. TradeLayer enable's the economies of Bitcoin-like blockchains to sustain decentralized banking and trading system independent from external price information and bank accounts.

1. Introduction

Commerce using Bitcoin has come to rely almost exclusively on financial institutions serving as trusted third parties to process cryptocurrency withdrawals. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust-based model.[1]

In order to make Bitcoin and its adjuvants a more perfect money, we must solve the key problems of decentralized exchange latency, make custody easier for individuals to manage, and create liquid markets in on-chain derivatives to enable native currency in users' chosen denominations.

There are five threads of research and development that came together to enable the protocol alternative presented in this paper: decentralized derivatives work started in 2014 by this author and extended by this author's colleagues in 2017/18, the Tendermint protocol [2],

general potential designs for OmniLayer [3] style transactions embedded into OP_Return payloads on Bitcoin-like blockchains, State Channels, and SPV Proofs[6].

The derivatives clearing algorithms demonstrate the level of useful complexity that can be implied by an OP_Return transaction, in this case allowing for leveraged trading and path-dependent bilateral settlement of contracts. The algorithm will receive treatment in its own yellowpaper.

The introduction of leverage enables the issuance of currency. Currency historically is not power money, it is a derivative of money, a B-side, a receipt. Since the loss of the gold standard, the US dollar has been currency used to purchase oil. In the 21st century, redeemable currency backed by various kinds of cryptocurrency will compete with bank deposits. These decentralized currency units can be used as collateral for on-chain peer-to-peer derivatives trades, or vouch to a sidechain and act as a validator, such as a sidechain staked under the Tendermint protocol as an alternative to the Cosmos Atom currency created by Tendermint's developers. It eliminates or dependence on issued bank account receipts, such as Tether USD, but it also complements such regulated currency tokens as they would generally pay less yield for holding them, but also carry regulatory guarantees the market may find attractive.

The Tendermint sidechains enable honest State Channels to relay state-data back to the Bitcoin-like home blockchain, which replaces the role of the Cosmos Hub as a root. SPV Proofs are used to insure against block reorg attacks on the home chains. Unconfirmed payments are temporarily underwritten by the pool of capital backing the insurance sidechain, which records the transactions and commits to relaying them again.

The sum of these functions is a decentralized exchange environment capable of:

1) pairing the home chain's native currency with any token on the layer (the native metacoin, any hedged currency, issued securities)

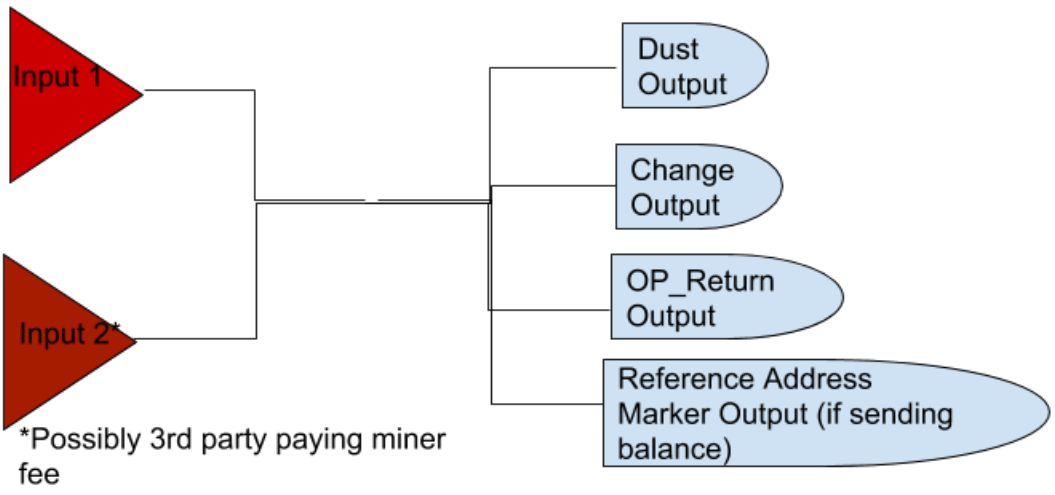
2) pairing any two tokens on the layer

3) trading decentralized derivatives that settle based on all of the above on-chain data

4) providing one second order execution and cost-effective order cancellations/amends

2. Transactions

Transaction inputs consist of one or more user-provided inputs, and an optional external input from a service provider. Transaction outputs consist of: 1) a change address, 2) reference address with dust amount (optional to tx where external address is the target), 3) an OP_Return. OP_Return payloads are base58 encoded strings that, when parsed, segment into the following fields: 2 bytes -> protocol marker, 2 bytes -> tx type, n bytes -> param1, ect. The OP_Return maximum size allocation varies between Bitcoin-like blockchains. Large and rare transaction types can be encoded into concatenated series of tx emissions that all reach the mempool in sequential order, such as may be necessary to broadcast proofs. OP_Return merkle branch continuity is parsed to arrive at a consensus hash about the state of the layer's balances, orderbooks and so on, which enables checksum handshakes between nodes.



An input is needed to publish the transaction and produce a dust output, the dust threshold is a low number of units that must be spent to put a transaction on the Bitcoin blockchain. A 3rd party service provider could fund the miner fee component of the tx, cover in detail below. Earlier dust inputs could be recycled for new dust outputs if fee difference is covered externally. If not then the user will want a change address to keep the surplus amounts from the inputs. Having published a dust output, the user is allowed one OP_Return output to encode a layer transaction payload. If sending or granting tokens, a tx may also color a reference address with another output, or for a single send the dust output serves this purpose.

3. Timestamp Service

Timestamping trading orders according to the mempool and block confirmation timestamping of the Bitcoin protocol is problematic. On the miner-side, any miner/node/trader can refuse to relay orders and broadcast their own instead, the cost of such a “relay attack “ is trivial. On the node side, percolation time of newly broadcast transactions to be timestamped for the mempool is unpredictable. In modifying the design of the OmniLayer protocol, which depended on mempool timestamps for order matching, mempool timestamps are used more extensively in TradeLayer, but generally as contingent logical fallbacks deferring to reference transactions published by State Channels that are governed by sidechain validator pools.

The Tendermint-based order matching allows cancels and amends to be low or no cost (depending on a pool’s public parameters), and occurring at a throughput of over 10,000 adjustments or removals per second. Tendermint utilizes a round robin order of block propositioning, so that for N validators every N seconds, a validator has the opportunity to order that second’s block in whatever order they please, and would use this monopoly on that atom of time to earn market maker rebates for having their matching orders come first in the block. We defeat ultra-high-frequency trading by defining the incremental monopoly on that unit of time with money-at-stake that each market-maker/validator contributed.

4. Proof-of-Byzantine-Failure

The strengths of the Bitcoin and Tendermint protocols complement each other, while ameliorating their respective weaknesses.

It’s possible to secure capital pledged for a sidechain by publishing an on-chain claim transaction. The initial pledge of capital comes from a Voucher transaction type that reserves a balance property on the layer and allows the sidechain validators to accept it as a deposit and welcome a new validator. Together the validators sign blocks and can include “slashing” penalties for those who do not follow the protocol, which includes responsibilities to publish reference transactions broadcast from the state channel address on the home chain. Common pubkeys used to generate the multisig address as used to generate the sidechain.

The state channel periodically publishes a merkilized meta-blockheader summarizing a set of blocks, these check-ins can be infrequent (e.g. 100 to 4000 block intervals) because validators all accumulate signed, nonce-timestamped transactions that *could* be broadcast from the State Channel if necessary but generally are held as insurance to save on cost. An SPV Proof allows a claimant to show they have continuity between the block they are making a claim about and the last state of the sidechain as visible on the home chain. The model of accumulating “cheques” in the form of signed, nonced transactions for each new sidechain block, means SPV proofs only have to connect one block to the following, allegedly Byzantine

block, this allows for very data-concise proofs, possible thanks to the single-block finality of Tendermint.

The second component of a claim is then presenting the state data about the block one is claiming was signed by a 2/3rd majority of Byzantine Actors. The signatories are judged to be Byzantine if the block they agreed to sign is improperly formed, exemplified by attempts to slash (punish monetarily) other validators in a non-standard way, transactions that don't have sufficient funding, or that violate the trading parameters advertised by the sidechain's validator pool since its formation.

The functions to define what is Byzantine or not will be hardcoded into the home blockchain-side client and the associated layer protocol consensus, a more modular re-factor of the way the layer protocol reasons about state could potentially enable more dynamic app integration with numerous variations on the audit function and associated app behavior.

5. Network

The steps to operate a TradeLayer Sidechain are:

1) Initialize bundled client, sync home blockchain.
2) Initialize sidechain with public keys corresponding to those used to create the multisignature 80% of M administrative address and the 1 of N broadcast address. Each single-signature address corresponding to those public keys must broadcast a transaction entering the validator pool formally, in order to receive an inflation subsidy from the protocol.
3) Wallet apps scan for and handshake with validator pools.
4) Transactions are multicast to both general node network and validator nodes.
5) Validators quickly order transactions into sidechain blocks.
6) Co-Validators sign proposed block after running checks on validity.
7) Block proposers continue the round robin cycle while scanning the home. blockchain for non-affiliated orders on the same pair.
8) If a match occurs on the sidechain based on fast amendments to orders that otherwise haven't visibly moved on the home chain, a reference transaction indicating a fill or partial-fill will be broadcast from a 1 of N multisig broadcast address by the next block proposer in the round robin.
9) If an amend has occured on the sidechain and a would be unaffiliate order enters that mempool that could match it, a reference transaction indicating a cancel or replace would be sent from the broadcast address.

Traditional OmniLayer consensus normally considered order matches based on the mempool timestamp as held by node consensus at the a new block containing the first confirmation is published., the time between broadcast and mempool inclusion varies based on

unpredictable global percolation between many nodes. In TradeLayer, a confirmed transaction effects a match if no reference transaction is included in the mempool before the timestamp of the confirming block. In this manner network congestion won't break the validators' ability to relay data. Late reference broadcasts, or failing to propose blocks, incur mild slashing penalties for the validators whose turn it is but who fail to act, while overt Byzantine actions incur more serious penalties.

6. Incentives

The original Master Protocol/Mastercoin, re-branded to Omni Layer/OMNI, had a flat supply bestowed to the original investors, plus a 10% vesting component that was meant to fund development but as of mid-2018 has not yet been deployed. Full validating nodes in the Bitcoin-family protocols do not earn any revenue for their contribution to network resiliency either. Centralized exchanges in unregulated jurisdictions have an incentive to wash trade at no true cost to themselves, to pump volume, and by correlation, revenue. The meta-coins ALL and TOTAL for the Litecoin and Bitcoin versions of TradeLayer respectively, exist as cousins to OMNI, with dynamic emission patterns. The core use case for the meta-coin is to accrue trading fees as a deflationary force, to incentivize behavior as an inflationary force, and to allow a float of stable currency based on derivatives of the meta-coin. Inflation in the meta-coin occurs in the following ways:

1) Node operators relay transactions periodically including a cipher about the latest block's resulting TradeLayer consensus hash, at periodic intervals based on the last characters in the new blockheader as it corresponds to the last characters in the user's address, such that a check-in occurs every twelve to twenty four hours, but in an unpredictable pattern. The fastest bunch of mempool timestamps with the correct ciphers earn each subsequent block's inflation reward for node operators.
2) Market liquidity providers earn the meta-coin as a secondary rebate for having their orders matched by someone else.
3) Social referral tokens are generated at the expense of some of the meta-coin, and earn the creator a reward based on the trading volume over time of referred addresses.
4) A set of addresses holding vesting tokens lock in a reserve balance of Founder Reward that unfreezes as the cumulative volume of the decentralized exchange increases. The units vest in a slowly accelerating sigmoid curve along the axis of cumulative trading volume. The total vesting sum will be roughly equal to 4.5% of the total supply at maturity, in line with the fully diluted percentage of the bitcoin money supply that Satoshi held from early mining.

Trading will initially incur the following fees:

Token-to-Token: Maker, -0.03%; Taker, 0.05%

UXTO Native Coin-to-Token: Maker, -0.025%; Taker, 0.05%

On-chain Data Settled Contracts: Maker, -0.01%; Taker, 0.02%

Oracle Settled Contracts: Maker, -0.01%; Taker, 0.025% (0.0075% accrues to Oracle)

The taker fees and rebate ratios for the pairs that create settlement data for contracts are more sharp to create a concave risk profile for a would-be market manipulator [7] having to pay taker fees and contend with the liquidity of all the properly incentivized arbitrageurs picking up multiple basis points of rebate income.

To retain long-term pricing power against competing exchange models, the fees can scale downwards algorithmically. Every order of magnitude increase in fee revenue above a de minimus threshold can trigger an N% decrease in fees, triggered every Y blocks (somewhere as often as 6 months or as infrequent as two years).

The primary focus does not need to be on people buying the meta-coin, but on hedging it into a synthetic version of the native currency and on synthetic versions of popular fiat currencies (e.g. sLTC, sBTC, dUSD, dJPY, dEUR et al.). Speed up trading in UXTO coins for layer tokens through sidechains, and the early use of oracle-settled contracts, greatly ameliorates the cold start problem around data manipulability for settlement, and liquidity risk trading from native coin to layer assets.

As more traders hold the meta-coins, the effectiveness of the “house” potentially wash trading for its own aggrandizement becomes infeasible. The cartelization we’ve seen traditionally with large exchanges, and again with cryptocurrency exchanges, can be diluted by the increasingly broader base of holding in the beneficiary asset. However Wash Trading remains a pattern attack vector, essentially a Sybil Attack, to earn these rewards from less-than-bona-fide activity. The order-sorting mechanism of a sidechain could allow pairs of validators to organize wash trades between each other easily, guaranteeing the specific matches they want. To combat this, a graph-parsing algorithm will be designed into the proof mechanism so that a whistleblowing validator can ask the protocol to verify this behavior. Single signature transactions broadcast from the 1 of M broadcast address are generally attributable to the individual validators, hence pattern Wash Trading can be detected. Only a sidechain that is 100% cartelized ever has a chance of sustaining cartel-like behavior profitably, and the free market should be able to favor more clean-looking performance histories from more legitimately diverse validator pools.

7. Reclaiming UTXO Space

Bitcoin-protocol design practitioners have a civil duty to design for data-efficiency where possible to no incur a tragedy of the commons. We can't go to 0 dust thresholds because the potential spam of mostly friction-less OP_Return strings could fill up the backlog for practically no cost. We could go to lower dust thresholds and more importantly larger OP_Returns, such that multi-state referencing transactions such as Send-To-Many are doable with fewer 100 byte signatures. Dust outputs however, all require their own 100 byte signatures to clean up and recycle them into future transactions, crowding the UXTO set and heightening RAM requirements to run a client. Prioritizing that clean-up by inserting the dusts into unsuspecting wallet users tx's is bad user experience design. Independent Segregated Witness transactions, or under an alternative mechanism that helps reduce the cost of multi-input transactions, should probably be the industrial-best-practices way of periodically cleaning up large numbers of dusts with minimal cost.

8. Insured Payment Verification

A Tendermint app supported by the correct proofs can also serve to timestamp zero-confirmation transactions in addition to timestamping orders; this can serve either for direct UXTO payments for a merchant use-case, or UXTO payments for token trades. The Simplified Payment Verification section of the Bitcoin Whitepaper didn't address the possibility of selfish mining attacks triggering a block reorganization based on a new longest chain emerging. While the risk of these attacks seems unlikely in general, and trivial in the case of small 0-confirmation payments, it becomes manifest when considering decentralized exchange transactions worth millions of dollars where one side is transacting with a UXTO payment. While tokens will never be released to a buyer paying with a UXTO until at least 1 confirmation has been read by the protocol, trades themselves have economic value beyond just return-of-principle, it should be possible to get a lock on price quickly.

Because a valid, single-signature raw transaction string is a high entropy string, the Insured Payment Verifiers (IPVs) who are in possession of that string can relay the transaction in the event that it is sent to them without the sender broadcasting it to the general node network. Then proofs must ensure that the validators are not corrupt and will indeed re-relay it in the event of a block reorg. Additionally, double-spend attempts on the part of the user can be insured against by requiring that the transactions use replace-by-fee as a template for its construction, so the re-relayers can thwart a double spend by broadcasting the original transaction with a higher fee. IPVs are compensated in layer property with a Send-to-Many transaction for payments, with the sold token in the case of token trades, and with an additional output of the native UXTO coin in the case of UXTO to token buys.

There are five scenarios to proof against, the positive, the negative, the false negative, the false positive, and the static situation where there is nothing to prove. The false negative

scenario is proofed against by the insurance app itself, any payment that was insured in the sidechain and no longer shows up in the home chain can be re-broadcast. The false positive scenario is to ensure that nobody can use a proof mechanism against honest validators by falsely asserting missing payments, the ability to claim a Byzantine failure as, for example, blocks are added to the sidechain containing payments that haven't reached the home chain's mempool, protects honest validators against a truly successful false positive.

The negative scenario is that a block reorg occurs and the culprits, somehow with significant cartel power as traders, miners and validators simultaneously, have colluded to not re-relay the insured transaction; in this scenario one honest validator can act as whistleblower and prove their case by re-relaying the transaction and having it reach the mempool as a valid one.

The positive scenario is, how do validators, in perhaps the most literal sense of the phrase yet used - virtue signal - that they are on board with the re-relay, when a truthful fulfillment of the insurance is carried out. They can broadcast a new block header at their admin addresses that is not contested by another validator as being false, the risk of proofs being used by whistleblowers is a major factor in that consideration, and then ensure that nobody needs to publish elaborate proofs to track back to the last check-in that preceded the block reorg period.

Finally when there is nothing to prove, any claims that there has been a failure to record a transaction can be dealt with against the last check-in of that sidechain's history, and disputes around those check-ins are dealt with along the N block confirmation "friendly finality gadget" of accepting a sidechain check-in as having finality on a blockchain like Bitcoin's. Additionally having validators "caching cheques" by retaining signed single-block header tx's potentially allows for disputes to be presented readily based on Tendermint's finality even if there is a very large re-org on the home chain.

This is still very experimental at the time of publication. However a key distinction between this approach an other sidechain projects including research done by Rootstock Labs [4] and IOHK [5] is the reliance of a single-block finality protocol for the sidechain and an avoidance on the compound headaches of reconciling state with some sense of finality between two pseudo-finality protocols.

9. Combining and Splitting 3rd Party Fee Payments

Users must spend the native UXTO coin in the form of miner fees to be able to publish transactions to the home chain. In order to open the user experience up to less experienced people, third party services can provide either a predecessor input or a secondary input to transactions, based on custody-less wallet integration. In the case where one has dusts on an address to choose as inputs, a secondary input to mine can top-up the BTC/LTC/BCH needed to pay the miner fee for that transaction, with a Send-to-Many OP_Return transaction sending payment both to the original destination, and a small payment in a balance property (such as

dUSD) to cover the cost of the miner fee with a competitive markup. In cases where there are no UXTO inputs on an address, the following procedure makes a transaction possible:

1) The wallet scans available validator pools that support this service and pings one.
2) The validator service spins up an unbroadcast txid for a payment of small quantity to the user's address, devises a subsequent raw tx string based on it, the necessary references and the OP_Return send-to-many, and shares that string with the wallet.
3) The wallet verifies that form of the transaction, signs it and pings back the resulting signed raw tx string to the service.
4) The service then broadcasts its funding tx, then the payment tx.

Since the second tx is invalid without the first, and because the wallet does not get the predecessor txid to potentially broadcast unilaterally, no party can defraud the other in this process, which can take two to three seconds to complete.

10. Custodial Security

There are two main security techniques needed to make cryptocurrency secure for most people, who are generally exposed to the risk of physical duress over bearer assets. The first technique is segregating assets over time, a Savings Address feature allows rate-limits, time-locks and cancelable cheque payments that ameliorate the risk of non-physical compromise (i.e. 'hacking').

The second technique involves segregating control over the assets with multisig back-up key holders; traditionally such a strategy only works for people hiring custodial trustees or hiding hand-written keys in miscellaneous forest dig sites and bank deposit boxes. To make safe custody truly scalable, the cost of customer service for modularly factoring out multisig custody must approach a combination commission-based finder's fee for lost keys' assets, and a low fixed annual fee. A Tendermint app where trustees have capital pledged and can be held to account if they sign non-compliant transactions, brings the cost down.

In a physical duress situation a pure Savings Address defense makes violence the high probability strategy for the assailant, to prevent the holder from canceling the transaction. Whereas, a 3rd party service needing secondary verification within twenty minutes to not cancel the payment, only operating during daytime business hours, changes the game theory to favor faster, less violent shakedowns for loose single-signature pockets of funds. It's the combination of the two that solves custodial security.

11. Calculations

Max Negative Yield based on Borrow Rates

0 = (Fi(x) - (BTC Borrow Rate + Metacoin Borrow Rate + Fees))-Risk-free Rate

Fi is a function that takes the average discount or premium of a perpetual swap contract, x, and returns an annualized rate of return based on the resulting swap payments. This income calculation function may be complex pending further research, and is truncated here for simplicity. If the swap longs can borrow and sell-short at a certain rate, they must receive an income rate that will pay a net-return at least as large as the prevailing risk free rates available in the derivatives markets.

12. Conclusion

While on one hand many of the problems Bitcoin sought to solve have yet to be solved, it's encouraging that the size, intellectual heft and temerity of the open source scientific movement it launched has given us enough shoulders of giants to climb over in presenting the solutions described in this paper. It should not be understated, it is amazing unto itself that the economic engine of Bitcoin has survived and reached centi-billion dollar valuations over 10 years.

Bitcoin succeeded because it combined four novel R&D threads from various predecessors, and leveraged the popular seigniorage of the block reward to encourage more network value than diluted through inflation (arguably, orders of magnitude more value). The same is going to be true for decentralizing the liquidity around these blockchains.

Tendermint's approach to Byzantine Fault Tolerance was a major breakthrough, distilling incentives and game theory into well-defined strengths and weaknesses. OmniLayer's approach to innovation on Bitcoin enabled a form of money that could be interactive with Tendermint, and interactive with native derivatives. The complex/high-throughput state machines of Tendermint, combined with collateral working in the layer consensus logic, resting on the security of proof of work, enables the best of both worlds. SPV proofs enable arbitration on the main chain over sidechains, and State Channel caching of signed blockheader transactions compliments the tail-risk of blockchains with pseudo-finality. The infrastructure enables low latency and security in a global, distributed, adversarial environment, which enables liquidity, and tools to manage private key custody risk make forms of money based on that liquidity usable for the average person.

Peer-to-peer electronic cash is a basic human right. Human beings should not have to sign an unfavorable legal contract in order to have the ability to transact in the stable unit of account most popular in their home region. We can give decentralized banking to billions of human individuals who most profoundly deserve it.

References

[1] Bitcoin Whitepaper
https://bitcoin.org/bitcoin.pdf

[2] Tendermint Whitepaper
https://tendermint.com/static/docs/tendermint.pdf

[3] OmniLayer Reference Spec
https://github.com/OmniLayer/spec

[4] Sergio Lerner, Leaf Node Weakness in Bitcoin Merkle Tree Design
https://bitslog.wordpress.com/2018/06/09/leaf-node-weakness-in-bitcoin-merkle-tree-design/

[5] Non-interactive Proofs-of-Proof-of-Work
https://eprint.iacr.org/2017/963.pdf

[6] Difference Between Sidechains and State Channels
https://hackernoon.com/difference-between-sidechains-and-state-channels-2f5dfbd10707

[7] Futures Manipulation with “Cash Settlement”
https://sci-hub.tw/10.1111/j.1540-6261.1992.tb04666.x