The bxHive Protocol

Authenticating Social Science Experiments

bxHive Team[1]

Wednesday 24th June, 2026

Abstract

Published results from the social and behavioral sciences often cannot be reproduced. This slows scientific progress and reduces confidence in scientific claims. Solving this problem is particularly difficult for social science experiments because subjects make decisions within decision environments that are often hard to fully record and authenticate. A transfer in a trust game, a bid in an auction, or a contribution in a public-goods game is not self-interpreting. It becomes evidence only when we understand the circumstances underlying the subjects' choices. The bxHive protocol authenticates the record of an experiment by providing a chain of custody among the six records required for reproducibility. These records are: the research question, experimental design, experimental methods, subject interface, decision setting, and outcome record. The bxHive application implements the protocol on the Algorand blockchain using the trust game as an example.

Keywords reproducibility, open science, microeconomic systems, bxHive Protocol

[1]Correspondence to: admin@bxhive.org

1 Introduction

1.1 The Reproducibility Problem In Social Science Experiments

A study of 600 published papers found that many results could not be independently reproduced because data or code were unavailable. Even when these materials were available, researchers were not always able to recover the published results [Miske and others, 2026]. Earlier large-scale investigations in psychology, experimental economics, and the social sciences, published in Nature and Science, have reported related concerns about replication and reproducibility [Open Science Collaboration, 2015, Camerer et al., 2016, 2018]. Together, these studies suggest that the reproducibility problem extends beyond individual papers and points to broader challenges in how scientific research is organized and documented.

Reproducibility is important because science advances when researchers can reproduce prior results and agree on what they see. This makes it easier to compare studies, find errors, replicate promising findings, and extend useful research programs. Reproducibility also matters outside the research community. Social scientists study institutions, markets, organizations, and public policy. When the record behind a finding is unavailable or incomplete, researchers have less ability to evaluate the claim, and the public has fewer reasons to trust the scientific process.

Human-subject experiments face a particular problem for reproducibility because they are designed to examine the decisions of people in institutions operating in a specific social environment. An institution defines the rules under which people interact. The environment defines the possible states of the world, who participates, and the technology used to change states. In an experiment an initial state is set and participants send messages to the institution. The rules of the institution then use the messages to change state. Participants are assigned preferences over possible states and are incentivized to choose messages to reach more preferred states.

The environment describes the characteristics of participants including what they know, what resources they control, their preferences over states, and their prior histories and beliefs. Experimenters use experimental controls where possible and randomization where they cannot. The data produced by an experiment are therefore not independent of the details of the institution and environment. If other researchers cannot identify these details they may see the same dataset without agreeing on what the observations mean.

Consider the trust game introduced by Berg, Dickhaut, and McCabe [Berg et al., 1995]. One participant receives an endowment and decides how much to send to another participant. The amount sent is multiplied, and the second participant decides how much to return. A transfer of ten dollars in this setting is not merely a data point. Its meaning depends on the rules of the game, the multiplier, the information available to participants, the payment procedures, and the sequence of decisions. Change any of these conditions, and the same transfer may reflect a different form of behavior. To understand what the data show, researchers must be able to observe the institution and social environment that produced them.

1.2 The Six Parts of a Reproducible Experiment

The trust-game example identifies the central problem of experimental accountability. To reproduce a human-subject experiment, later researchers need access to six parts of the experimental record.

1. The research question, which states the problem the experiment is designed to study.
2. The experimental design, which states the treatments, assignment rules, sample size, and planned comparisons.
3. The experimental methods, which include the procedures, data construction, exclusion rules, statistical models, code, and analysis workflow.
4. The subject interface, which records what subjects were told, what they saw, and how they submitted decisions.
5. The decision setting, which defines the environment and the institution including roles, allowable decisions, information, state transitions, outcomes, and payment rules.
6. The outcomes record, which records subject decisions, state changes, earnings, payments, constructed datasets, and reported results.

Together, these records document how an experimental result was produced and provide the basis for independent evaluation and reproduction.

Having these records is not enough. Researchers must also know whether the records are authentic and how they are connected. A research question should be linked to the design that was used to study it. The design should be linked to the methods, interface, and decision setting that implemented it. Subject decisions should be linked to the states, payments, datasets, and results they produced. Reproducibility, therefore, requires more than access to separate research files. It requires an authenticated chain of experimental records. The bxHive protocol is designed to create that chain.

1.3 What is the bxHive Protocol?

The bxHive protocol is an open-science protocol for authenticating social science experiments. It is designed to create an authenticated chain linking the six records that make up the experimental record. The protocol uses cryptographic commitments, signed messages, smart contracts, and linked artifact records to make the experimental record harder to alter and easier to evaluate. bxHive makes the experimental record more complete, more durable, and more accountable. The protocol is independent of any particular blockchain, but the first bxHive application implements these ideas on the Algorand blockchain.

The bxHive application implements the protocol on Algorand. Algorand is used because it supports signed transactions, smart contracts, low-cost state changes, and settlement of subject payments. The application starts with the trust game because it is simple enough to implement clearly, but rich enough to demonstrate the six records required for reproducibility and the chain of custody that connects them. An experimenter can preregister a study, configure the trust-game decision setting, fund escrow, open the experiment, and record the resulting decisions and outcomes. Subjects enter through a web interface, authorize decisions through signed transactions, and receive payments according to the rules implemented by the smart contract. This application demonstrates how bxHive can authenticate and establish a chain of custody among the six parts of the experimental record.

The existence of an authenticated experimental record does not by itself explain what should be included in it. The six records identified above are derived from the broader open-science requirements concerning reproducibility, transparency, accountability, and scientific reuse. The next section examines these requirements and shows how they were used to design the bxHive protocol.

2 Open Science Requirements for Human-Subject Experiments

Open science is often proposed as a solution to the reproducibility problem. For human-subject experiments, however, reproducibility depends on maintaining an authenticated experimental record. The six-part record introduced above defines bxHive's design target. In this section, we specify the open-science requirements that the record must satisfy.

A reproducible experiment begins before the experiment is run. Researchers make decisions about the question they wish to study, the design they plan to implement, the methods they will use, the interface subjects will encounter, and the decision setting in which subjects will act. Recording these commitments before observing outcomes helps distinguish planned tests from later exploration and provides a reference point for evaluating reported results. For this reason, preregistration anchors the first five parts of the experimental record. Without these commitments, later researchers may be able to reproduce a statistical result without knowing whether it answers the question the experiment was originally designed to investigate.

While preregistration records what was planned, provenance records what happened after the experiment began. For human-subject experiments, this path matters because data are produced through a chain of events. A reproducible experiment requires a chain of custody linking the preregistered records to the outcome record. Without this chain of custody, later researchers may know what was planned and have the final data, yet still be unable to determine what experiment was actually run.

After an experiment is run, the outcome record is converted into a statistical result. A researcher decides how to construct the dataset, which observations to exclude, which variables to define, and which statistical model to estimate. For this reason, reproducible analysis requires a record of the path from the outcome record to reported findings. Later researchers should be able to begin with the outcome record and recover the tables, figures, estimates, and tests reported by the authors. If that path is missing, researchers may see the same outcome record but disagree about how the reported finding was produced.

In human-subject experiments, the record often includes consent forms, subject identifiers, payment information, and behavior that may become sensitive when linked across studies. These materials still matter for reproducibility, but access to them has to be governed. Later researchers need to know that a record exists, who is authorized to inspect it, and what conditions apply to reuse. The open-science requirement is therefore controlled access to an authenticated record, not unrestricted disclosure.

The requirements discussed above define the information needed to evaluate, reproduce, and extend a human-subject experiment. Preregistration records what researchers planned to do. Provenance records what happened after the experiment began. Reproducible methods record how reported findings were derived from the outcome record. Controlled access governs the sharing and reuse of sensitive records. Together, these requirements define the six-part experimental record introduced in Section 1. The bxHive protocol was designed to authenticate these records and maintain the links among them.

3 The bxHive Protocol

The objective of the bxHive protocol is to authenticate the experimental record of a human-subject experiment. An authenticated record is not just a collection of files. It is a chain of custody linking the preregistered records to the outcome record produced by the experiment. The chain begins with the research question, experimental design, methods, subject interface, and decision setting. It then records how subjects entered that setting, what decisions they made, what outcomes followed, and how those outcomes were converted into reported findings. In this way, bxHive allows later researchers to determine what was planned, what was implemented, what decisions were made, and how reported findings were produced.

The bxHive protocol uses a three-layer architecture to manage different parts of the experimental record.

1. The registry layer authenticates the experimenters and subjects responsible for creating and executing these records.
2. The experiment layer handles what study is being run by recording the research question, experimental design, methods, preregistered commitments, treatment configurations, and subject interface.
3. The institution layer handles how outcomes are produced by implementing the decision setting, validating subject decisions, updating experimental states, calculating earnings, and recording resulting outcomes.

Together, the layers create the chain of custody linking the preregistered records to the outcome record.

3.1 The Registry Layer

The registry layer authenticates the experimenters and subjects responsible for creating and executing these records. It stores on-chain records for experimenters, who set up and run experiments, subjects, who participate in a session, and platform administrators, who handle registry access and approve experiment templates. Each account gets a role and a unique ID that stays the same across experiments. When an experimenter starts a study or a subject joins a session, the protocol checks the registry to ensure the account is authorized. Since every action needs a registry check, each event in the experiment record is linked to a verified account. By separating participant records from participation records, the registry layer allows the same subjects and experimenters to participate across multiple studies while maintaining an authenticated record of their participation.

3.2 The Experiment Layer

The experiment layer is a smart contract that records the experimental design for a study. It is populated when the experimenter completes the experimental design template in the bxHive web application. The contract stores the treatment variations, subject assignment rules, number of institution contracts, and subject interface. When the contract is created, it preregisters the experimental design. When the experimenter sends a create_variation() message, the experiment layer creates the institution contracts specified in the design. The experiment layer contract ensures that the preregistered design is used.

3.3 The Institution Layer

The institution layer is a smart contract that implements the decision setting for a treatment variation. The experimenter creates one institution contract per treatment variation and funds it with escrow. The contract enrolls subjects and pairs them into matches. Each match runs through a sequence of decision phases defined by the institution layer. The contract validates each decision against the institution's rules before accepting it. When the final decision in a match is submitted, the contract calculates earnings and transfers payment directly from escrow to each subject. All matches for a variation run inside the same contract, so the institution layer holds the complete record of decisions, outcomes, and payments for that variation.

3.4 The bxHive Website and User Wallets

The bxHive website serves as the main user interface for the protocol. Participants sign in using their wallet, and all transactions sent to the smart contracts are signed by that wallet. Experimenters use it to register accounts, set up experiments, configure treatments, preregister designs, and monitor sessions.

Subjects use the site to sign up, join experiments, read instructions, make decisions, and check their earnings. Administrators use it to manage registry access and approve experiment templates.

The website allows participants to send transactions to the smart contracts across all three layers of the system. It communicates with the registry layer when accounts are created and permissions are set, with the experiment layer when studies are designed, configured, and started, and with the institution layer when subjects participate in sessions. The website is not part of the experimental institution. It provides access to the records and decision settings that the protocol manages.

3.5 Authentication and Provenance

Authentication uses a blockchain's cryptographic properties. Every transaction submitted through the website is signed by the participant's wallet. The signature identifies who authorized the transaction and cannot be forged or easily altered. The blockchain timestamps each transaction when it is recorded, establishing when each event occurred. Off-chain materials, such as the web interface used and the statistical code used to analyze the data, are authenticated by keeping a hash of the file on-chain. If the file is later changed, the hash will not match, making any change detectable. Together, this approach allows other researchers to verify who did what, when they did it, and whether the materials they are examining are the same ones used when the experiment was run.

Section 1.2 listed six parts of the experimental record needed for a reproducible experiment: the research question, experimental design, experimental methods, subject interface, decision setting, and outcome record. Section 2 explained that reproducibility needs more than access to these records. There must be a chain of custody connecting them. The bxHive protocol builds this connection. The experiment layer begins that chain by recording the experimental design, experimental methods, and subject interface before the experiment begins. The institution layer continues the chain by implementing the decision setting and recording the outcomes produced by subjects. The registry layer authenticates the experimenters and subjects responsible for creating and executing these records. Together, the three layers allow future researchers to trace the experimental record and identify the authorized participants who produced it.

4 The bxHive Platform on Algorand

4.1 Why Algorand

The blockchain used to host bxHive must be able to run smart contracts and maintain an authenticated experimental record at low cost. bxHive needs signed transactions, smart contracts, low-cost state changes, reliable payment settlement, and durable storage for authenticated records. The protocol must also preserve commitments to designs, methods, interfaces, and other artifacts needed to reproduce the experiment. Algorand provides the blockchain functions needed to support this workflow, including accounts, signatures, application calls, programmable contracts, on-chain storage, fast finality, and transaction-level records [Gilad et al., 2017, Algorand Foundation, 2026].

4.2 Platform Architecture

The bxHive implementation relies on six Algorand components. Accounts identify experimenters and subjects. Smart contracts implement the rules of the protocol and the experimental institution. The contracts are written in Algorand Python using AlgoKit, which makes the code readable, auditable, and shareable as part of the experimental record. Box storage holds the protocol records that need to persist on-chain, including participant enrollments, match states, and payment records. Inner transactions allow the experiment layer to spawn institution layer contracts directly on-chain when an experimenter starts a study. Transactions carry signed messages between participants and contracts. Application state stores the data the protocol needs during execution. These components provide the foundation for authenticating records, executing decision settings, recording outcomes, and settling payments.

Algorand supports three network environments that correspond to different stages of development and deployment. LocalNet is a private single-node instance of the Algorand blockchain that runs on the experimenter's own machine. It is used to develop and test the protocol without incurring any costs and to enable classroom demonstrations of research experiments. Testnet is a public Algorand network that uses test tokens with no real value. It is used to verify that the protocol works correctly in a live network environment before real funds are involved. Mainnet is the live Algorand network where transactions are final, and payments use real ALGO or other Algorand assets. bxHive is designed to run on any of these three networks. These network choices allow researchers to develop and test experiments locally before deploying them to a live network.

4.3 bxHive Workflow

The bxHive protocol involves three distinct workflows: one for developers who build and deploy the protocol, one for experimenters who design and run studies, and one for subjects who participate in them.

The developer workflow starts by writing the smart contracts in Algorand Python using AlgoKit. Once written, the developer then builds the contracts to generate the TEAL bytecode and deploys them to LocalNet for testing. AlgoKit then generates TypeScript clients from the contract specifications, which the bxHive web application uses to communicate with the contracts. The developer builds and tests the web interface against LocalNet, where experiments can be run without cost. When the contracts and interface are working correctly on LocalNet, the developer deploys to Testnet to verify behavior on a live network before any real funds are involved. When the protocol is ready for production, the contracts are deployed to Mainnet. Any change to the contracts requires rebuilding, regenerating the TypeScript clients, and redeploying before the web application can use the updated contracts.

The experimenter workflow follows five steps. First, the experimenter registers an account with the registry layer through the bxHive website, establishing an authenticated identity on-chain. Second, the experimenter creates an experiment and completes the design template, which records the research question, treatment variations, assignment rules, and subject interface in the experiment layer contract. Third, the experimenter starts the experiment by sending a create_variation() transaction, which causes the experiment layer to spawn one institution layer contract per treatment variation. Fourth, the experimenter funds each institution contract with escrow, enrolls subjects, and pairs them into matches. Fifth, after the experiment runs, the experimenter exports the outcome record, which includes subject decisions, state changes, earnings, and payments, for analysis. At each step, the transaction is signed by the experimenter's wallet and timestamped by the blockchain, so the complete workflow is part of the authenticated experimental record.

The subject workflow follows four steps. First, the subject registers an account with the registry layer through the bxHive website, establishing an authenticated identity on-chain. Second, the subject reviews

the experiment instructions and enrolls in a session assigned by the experimenter. Third, the subject enters the decision setting, where the institution layer contract presents the decision problem, validates each decision the subject submits, and advances the match through its phases. Each decision is submitted as a signed transaction from the subject's wallet. Fourth, when the final decision in the match is submitted, the institution layer calculates the subject's earnings and transfers payment directly to the subject's wallet. The subject can then view their earnings through the bxHive website.

The next subsection describes how the records produced by these workflows are stored and how escrow and payments are managed.

4.4 Record Keeping and Subject Payments

The bxHive protocol keeps some records on-chain and others off-chain. On-chain records include account identities, roles, experimental designs, treatment setups, subject enrollments, match states, decisions, earnings, and payments. These are saved in the smart contract's application state and box storage, and each one is linked to the transaction that created it. Some records are too large to store on the blockchain, so instead, a hash of each file is saved on-chain when the file is added. This hash acts as a reference, helping to identify the file and making any later changes easy to spot. Together, the on-chain records and the hashed references to other files make up the full experimental record that bxHive verifies.

In bxHive, escrow and payments are handled by the institution-layer contract. Before the experiment runs, the experimenter deposits enough escrow into each institution contract to cover the maximum possible subject earnings for that treatment variation. The contract holds these funds until a match is complete. When the final decision is submitted, the contract computes earnings and transfers payment to each subject's wallet. Payment is therefore governed by the rule encoded in the contract rather than by an experimenter's later discretion. Once the contract is funded and the experiment is running, the experimenter cannot withhold or change subject payments. When all matches in a variation are complete, any unused escrow can be returned to the experimenter. This makes the payment rule part of the authenticated experimental record.

5 Trust Game MVP

The trust game introduced by Berg, Dickhaut, and McCabe [Berg et al., 1995] is the first experiment implemented in bxHive. The trust game is a two-player sequential extensive form game. The investor, DM1, receives an endowment E_1 and chooses an investment s, where $0 \leq s \leq E_1$. The amount s is multiplied by m and sent to the trustee, DM2, who receives an endowment E_2 and the multiplied amount $s \times m$. DM2 then chooses a return r, where $0 \leq r \leq s \times m$. The payoffs are:

$$\pi_1 = E_1 - s + r \tag{1}$$

$$\pi_2 = E_2 + (s \times m) - r \tag{2}$$

The trust game is a good starting point for two reasons. First, it is simple enough to implement clearly in a two-player institution with a small number of parameters. Second, it is rich enough to demonstrate all six records of a reproducible experiment and how bxHive authenticates each of them.

5.1 Experiment Workflow

The developer writes the institution layer smart contract to implement the trust game for general parameters E_1, E_2, m, and UNIT. The contract accepts any values for these parameters and enforces the rules of the game for whatever values the experimenter chooses. The developer also writes the experiment layer contract to store the experimental design and spawn institution layer contracts, and deploys both to the network.

The experimenter registers an account and completes the experimental design template in the bxHive website. For this experiment the experimenter sets $E_1 = 100$, $E_2 = 100$, and UNIT $= 10$, and specifies two treatment variations: $m = 3$ and $m = 4$. These amounts can be denominated in ALGO or USDC. The experimenter also sets the matching rule, pairing each investor with one trustee, and specifies 50 pairs per variation. The experimenter sends a create_variation() transaction to the experiment layer, which spawns one institution layer contract per variation. The experimenter then deposits escrow into each institution contract to cover the maximum possible earnings for all subjects.

When the experiment opens, each subject is matched with another subject and assigned a role. Consider Alice and Bob, one of the 50 pairs in the $m = 3$ variation. Alice is assigned the investor role and Bob the trustee role. In the experiment they are referred to as DM1 and DM2. Alice signs into the bxHive website using her wallet. She sees her endowment of 100 ALGO and the multiplier of 3. She invests $s = 70$ ALGO and submits a signed transaction. The institution contract validates the decision and advances the match to Bob's decision phase. Bob sees that Alice invested 70 ALGO and that he has received $70 \times 3 = 210$ ALGO. Bob returns $r = 100$ ALGO and submits a signed transaction. The institution contract validates the return, calculates the payoffs, and transfers payment to both wallets in the same transaction. Alice receives $\pi_1 = 100 - 70 + 100 = 130$ ALGO and Bob receives $\pi_2 = 100 + 210 - 100 = 210$ ALGO.

5.2 The Six Records in Practice

The trust game implementation produces all six records of a reproducible experiment.

The research question is entered by the experimenter in the design template and recorded in the experiment layer contract before the experiment runs.

The experimental design is stored in the experiment layer contract at the same time:

```
{
    experimenter_account: ALGO_account_address_0,
    signed_by: ALGO_account_address_0,
    timestamp: blockchain,
    E1: 100,
    E2: 100,
    UNIT: 10,
    variations: {m: 3, m: 4},
    pairs_per_variation: 50
}
```

The experimental methods, including the statistical code and analysis workflow, are hashed before the experiment runs and the hashes are stored on-chain, so any later change to those files is detectable.

The subject interface, which presented Alice with her endowment and multiplier and Bob with the amount received, is authenticated the same way.

The decision setting is implemented and enforced by the institution layer contract:

```
{
    DM1_role: investor,
    DM2_role: trustee,
    DM1_account: ALGO_account_address_1,
    DM2_account: ALGO_account_address_2
}
```

The outcomes record is produced by the institution layer as the experiment runs:

```
{
    DM1_decision: {
        account: ALGO_account_address_1,
        investment: 70,
        signed_by: ALGO_account_address_1,
        timestamp: blockchain
    },
    DM2_decision: {
        account: ALGO_account_address_2,
        return: 100,
        signed_by: ALGO_account_address_2,
        timestamp: blockchain
    },
    DM1_payout: 130,
    DM2_payout: 210
}
```

5.3 What the MVP Demonstrates

The trust game MVP demonstrates that the bxHive protocol can produce an authenticated chain of all six records of a reproducible human-subject experiment. The registry layer established verified identities for the experimenter, Alice (DM1), and Bob (DM2). The experiment layer preregistered the design, including the two treatment variations, before any subject decisions were made. The institution layer enforced the rules of the trust game, recorded DM1's and DM2's decisions, and settled their payments automatically from escrow without experimenter intervention. The website gave Alice and Bob access to the decision setting through a standard web interface, with each decision submitted as a signed wallet transaction. The result is an experimental record that a later researcher can examine to verify what was planned, what parameters governed the decision setting, what decisions DM1 and DM2 made, and how their payouts were calculated and transferred.

6 Conclusion

A study of 600 published papers found that many results could not be independently reproduced because data or code were unavailable, and even when those materials were available researchers were not always able to recover the published results [Miske and others, 2026]. Earlier investigations in psychology, experimental economics, and the social sciences reported related concerns about replication [Open Science Collaboration, 2015, Camerer et al., 2016, 2018]. These studies point to a common problem. The record behind a published finding is often incomplete, making it difficult for other researchers to evaluate the claim or for the public to trust the scientific process.

bxHive addresses this problem by authenticating the six records required for a reproducible human-subject experiment. Together, these six records form the authenticated experimental record. The registry layer establishes verified identities for everyone who acts in the record. The experiment layer preregisters the research question, experimental design, methods, and subject interface before the experiment runs. The institution layer implements the decision setting, validates subject decisions, records outcomes, and settles payments automatically from escrow. Authentication relies on wallet signatures, blockchain timestamps, and file hashes. Together, these mechanisms produce a connected record that runs from the preregistered design to the outcome record and that later researchers can examine to verify what was planned, what was implemented, and how the results were produced.

The goal of bxHive is not to solve the reproducibility problem by software alone. The goal is to make the experimental record more complete, more durable, and more accountable. A durable authenticated record supports more than replication. It supports the cumulative growth of scientific knowledge. Later researchers can replicate an experiment, modify a treatment, reanalyze the data, or build a new experiment from an existing design. Because these activities remain linked to earlier records, researchers can determine where a result came from and how it evolved.

bxHive is under active development. The current implementation is Milestone 1, a working trust game proof of concept on Algorand LocalNet. Milestone 1 provides architectural validation by demonstrating that a human-subject experiment can be configured, run, recorded, and recovered for analysis. Milestone 2, targeted for September 1, 2026 moves bxHive to Algorand TestNet and adds subject privacy and digital identity. The goal is a publishable laboratory experiment with a durable blockchain-linked record that protects human subjects by separating public provenance from private consent, eligibility, and payment records. Milestone 3, targeted for October 1, 2026 is a replication on Algorand MainNet with online subject participation through wallets. The goal is to show that bxHive can operate under a full consensus blockchain and support a replication that is explicitly linked to the original experiment in the authenticated record.

References

B. Miske and others. Investigating the reproducibility of the social and behavioural sciences. *Nature*, 652: 126–152, 2026.

Open Science Collaboration. Estimating the reproducibility of psychological science. *Science*, 349(6251): aac4716, 2015.

Colin F. Camerer, Anna Dreber, Eskil Forsell, Teck-Hua Ho, Jürgen Huber, Magnus Johannesson, Michael Kirchler, Johan Almenberg, Adam Altmejd, Taizan Chan, and others. Evaluating replicability of laboratory experiments in economics. *Science*, 351(6280):1433–1436, 2016.

Colin F. Camerer, Anna Dreber, Felix Holzmeister, Teck-Hua Ho, Jürgen Huber, Magnus Johannesson, Michael Kirchler, Gideon Nave, Brian A. Nosek, Thomas Pfeiffer, and others. Evaluating the replicability of social science experiments in *Nature* and *Science* between 2010 and 2015. *Nature Human Behaviour*, 2: 637–644, 2018.

Joyce Berg, John Dickhaut, and Kevin McCabe. Trust, reciprocity, and social history. *Games and Economic Behavior*, 10(1):122–142, 1995.

Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nichai Zeldovich. Algorand: Scaling Byzantine agreements for cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pages 51–68, 2017.

Algorand Foundation. Algorand Smart Contracts Overview. *Algorand Developer Documentation*, 2026.