Brave Ads x THEMIS RFC&C

Brave Research Team

February 2, 2021. Revision 1.0

Abstract Here we present the THEMISv2 protocol, a privacy-preserving, trustless, and verifiable protocol for calculating ad rewards in a decentralized setting. THEMISv2 was designed to be used as part of Brave Rewards, a privacy-preserving advertising system that incentivizes users to interact with ads in the Brave browser. Brave Rewards is widely deployed and used by many of the 24 M users of Brave. THEMISv2 preserves users privacy by design, allowing users to calculate their rewards and prove the computational correctness *without* having to disclose which ads they have interacted with. Moreover, THEMISv2 allows any participant to verify the correctness of the reward claimed by users. This technical report serves as the basis for the open Request For Comments and Code (RFC&C) event that starts in February 2021. The RFC&C event aims to encourage the research and crypto community to come together in order to comment, review, and to collaborate on the next-generation BAT ecosystem.

1 Introduction

The Basic Attention Token (BAT) [1] was introduced in a white paper mid-2017. Since then, BAT has been used by millions of users, advertisers, and publishers, all connected to the BAT ecosystem through the Brave browser.

The Brave browser and BAT both have experienced rapid growth in the last three years. Brave has over 24 million monthly active users as of December 2020, and BAT is one of the most widely-used tokens, with close to 12M wallets, 4.3M monthly transacting BAT users, over 1 million verified publishers, and more than 400 advertisers. To date, Brave ads and Brave rewards have displayed millions of opt-in ads to our users, while preserving the privacy of end-users and rewarding them for their interactions with ads.

While Brave honestly and fully executes its role as the guardian of this system, this comes with a downside: currently users and advertisers need to trust that the rewards being paid are accurately computed by Brave.

The Brave research team has been working on the THEMIS protocol [2] to progressively decentralize the BAT Ads and Rewards infrastructure, while providing users and advertisers with the ability to verify that the protocol is running correctly. Our goal is to design and implement a protocol in which advertisers (and Brave itself) can verify that the rewards claimed by users are

correctly computed, while keeping the ad interactions private. Additionally, we aim to ensure that the system can scale to millions of payout requests every month.

Ad-related interactions involve the end-user, their browser, and Brave servers. The browser downloads the whole ad catalog periodically and *locally* selects which ads to show to the user, as presented in Figure 1, i.e. outside of the main web page. The user then interacts with the ads (hovering over, clicking, dismissing every ad notification) and the browser informs Brave servers, through an anonymous channel, which interaction has been performed. The user receives a reward related to that particular event, which is anonymously redeemed together with all other rewards from other events at the end of the month.

However, the current protocol fails to provide verifiability of the rewards request, since the reward computation is computed by Brave. Currently, users and advertisers alike need to trust Brave that it is paying the correct amount of BAT to end-users.

This report outlines the goals and requirements for a decentralized, trustless, and privacy-preserving ad protocol (Section 2), to be used in the context of Brave Ads and Brave Rewards — the *target protocol*. In Section 3 we present the design of THEMISv2, a protocol that leverages cryptographic accumulators and zero-knowledge proofs to guarantee the goals and requirements outlined in Section 2. Section 4 describes how the RFC&C event is organized: the open challenges, timeline, submission details, etc. A live FAQ of the RFC&C is available on GitHub, where teams can ask questions. In addition, there is a public RFC&C technical Discord channel for discussions around THEMISv2 and the RFC&C event. Finally, in Section 5 we present an overview of some of the related work that should help the participants of the RFC&C event.

2 Decentralizing Rewards

In this section we describe the properties and guarantees the target protocol should have. RFC&C teams should keep in mind the goals and properties described in this section when working on improvements to THEMISv2 or when designing a new protocol.

2.1 Protocol Introduction

Goals

We aim to further decentralizing the current Brave Rewards system by designing and implementing a new protocol that we call the *target protocol*. More concretely, the target protocol should

- support reward computation based on user ad interactions without leaking information about user behavior;

Figure 1: Example of an ad notification delivered through the browser for Brave Ads users.

- allow all participants to verify that the rewards are being correctly computed; and
- allow advertisers to verify whether budget metrics (how the campaign budget has been spent) provided by Brave are correct.

The tension between the need for user privacy and the provable reward calculation is the main challenge of the RFC&C, together with the accompanying challenges of running Rewards at scale. In Section 2.2, we present an informal list of properties that we expect Brave Rewards to have in the long term. However, satisfying *all* the properties falls outside the scope of this RFC&C, and the teams may instead choose to focus on working suggestions that focus on specific properties.

Participants

The participants in the target protocol are the *users*, *advertisers*, *verifiers*, and *Brave*.

Users. *Users* interact with ads through the browser and keep the interaction state locally. At each interaction, they send confirmation events (specify-

ing the ad interaction) through an anonymous channel to *Brave*. Eventually, the *users* request a reward payment based on their interactions with ads over time. The reward request may be computed by *users* themselves (as long as they provide proof), but not necessarily; protocols may leverage a blockchain to calculate and verify the reward calculation.

Brave. *Brave* is responsible for validating the confirmation events. When it receives a new, anonymous, confirmation event of an ad interaction, it updates its local state and responds with a validation for the state update. This validation can take the form, for instance, of a cryptographic token.

Advertisers. Advertisers create ad campaigns with multiple ads which are distributed to users through the Brave Browser. For each ad campaign, *advertisers* and *Brave* agree on the ad campaign budget and how much a *user* should be rewarded for each interaction with an ad in the campaign. Every participant in the target protocol knows the reward amount of each ad interaction.

In the target protocol we can also consider the role of the *verifier* who is able to verify the proofs of correct reward computation on-chain or off-chain. Any of the participants listed above may perform the role of *verifier*.

Anonymous channel

The target protocol may assume the existence of a communication channel which allows the *users* to make requests to *Brave* without leaking their IP. The mechanism implementing the anonymous channel falls outside the scope of this report.

2.2 Properties and Requirements

The target protocol aims to fulfill the following informal properties:

Property 1: Privacy-preserving ad interaction: An ad interaction can not be linked to a *user*;

Property 2: Ad interaction unlinkability: Two ad interactions cannot be linked together by *Brave* or an *advertiser*;

Property 3: Advertiser campaign analytics privacy: The campaign analytics of an *advertiser* must be only available to Brave and the advertiser;

Property 4: Interaction state update verifiability: *Users* can verify that the interaction state is correctly updated during the protocol, according to their ad interactions;

Property 5: Decentralized reward request verifiability: Any participant can verify that rewards requests from *users* are valid with respect to the state updated by *Brave*; The result of the reward verification must be committed to a public blockchain for visibility purposes;

Property 6: Advertiser verifiability: *Advertisers* can verify that the budget expenses corresponding to their ad campaigns are being spent based on the confirmation events received by *Brave*.

One of the goals of the RFC&C is to design a target protocol that supports all or a significant subset of the above properties. As discussed, Property 1 may be achieved through the use of an anonymous channel. However, the target protocol must guarantee that no other component leaks sensitive *user* information. We should point out that while this is outside the scope of the RFC&C, a real-life deployment of such a protocol would include a fraud-prevention system; as part of that, we generally ensure that Brave is the only entity that is able to issue, update, and redeem tokens.

3 THEMISv2 and Its Building Blocks

In this section, we provide an overview of THEMISv2 and its building blocks, and how they interact to achieve the goals and properties outlined in Section 2.2. Figures 2 and 3 provide some illustrative flow diagram for THEMISv2.

3.1 Overview of THEMISv2

The THEMISv2 protocol allows *Brave* to distribute advertising budget allocated from ad campaigns by *advertisers* to *users* in the form of BAT rewards. THEMISv2 protocol guarantees that the rewards received by the users are correct, based on their interactions with ads. Additionally, *advertisers* can verify that their ad campaign budget is correctly used by *Brave*. To that end, the THEMISv2 protocol is divided in five distinct phases:

1. **Interaction state initialization**: *Brave* and the *user* initialise the state of the user's interactions with ads. The initial interaction state corresponds to 0 rewards (i.e., no ad interactions yet), and is encoded in a signed and tamper-proof cryptographic token (more details in Section 3.2).

2. **Interaction state update**: The *user* interacts with ads locally, and sends *Brave* its cryptographic token together with a notification of which interaction took place through an anonymous channel. *Brave* updates the token according to the interaction encoded in the request and sends it back to the user. The *user* verifies the correctness of the state update.

3. **Reward calculation**: Once the user has interacted with ads multiple times, it calculates the reward corresponding to the interaction state. Since i) the user knows how many rewards each ad pays and ii) the interaction state encodes all the interactions with the ads over time, the *user* is able to calculate the rewards locally. This way, the *user* is certain that the rewards were calculated as expected. In addition, the user proves that the reward calculation is correct based on the cryptographic token signed

by *Brave* given properties i) and ii). We give more details on how the proof is generated and verified in Section 3.2.

4. **Reward verification**: The proof, together with the reward result, the token, and the signature of the latter, are sent to *Brave*. *Brave* verifies the proofs, and makes them available to the *verifiers* by means of a public blockchain verification procedure[1].

5. **Anonymous and scalable payments**: Brave makes anonymous reward-related payments to users who provide a valid request.

3.2 Interaction State and Black-Box Accumulators (BBAs)

The interaction state keeps track of which ads a user has interacted with over time. The state is encoded as a vector, where each index of the vector represents how many interactions the user had with a particular ad.

To give an intuition on how the interaction state looks like, let's consider an example where there are five ads in the system, which the user must keep track of. The initial state is then represented by:

$$\texttt{int\_state} = [\texttt{int\_ad}_0, \texttt{int\_ad}_1, \texttt{int\_ad}_2, \texttt{int\_ad}_3, \texttt{int\_ad}_4] = [0, 0, 0, 0, 0]$$

If we consider that the user has interacted one time with $\texttt{int\_ad}_2$ and four times with $\texttt{int\_ad}_4$, the `int_state` should be:

$$\texttt{int\_state} = [0, 0, 1, 0, 4]$$

In the context of THEMISv2, the vector storing the interaction state is encoded as a cryptographic accumulator, called a black-box accumulator (BBA), introduced by Rager and Rupp [3].

Conceptually, BBAs are private counters that only their issuer can update[2]. They consist of the *state*, a hiding *commitment* of the state, and a *signature* over this commitment. The BBA can be randomized by the client without losing the integrity of the data structure, making two *show* events of a BBA effectively unlinkable. Moreover, the state of BBAs can remain hidden during the update, meaning that the issuer only knows its state at the time of initialization, when the state is at zero.

The protocol used by the *user* and *Brave* to initialize and update the interaction state using the BBA is shown in Figure 2:

① First, the *user* requests a new BBA from *Brave*;

② *Brave* issues and signs a new accumulator;

③ *Brave* sends the accumulator back to the *user*;

[1] Valid proofs from multiple users may be batched for on-chain verifiability.

[2] We give the technical details of the BBA scheme we use [4] in a different document that will be made available here, as well: github repo.

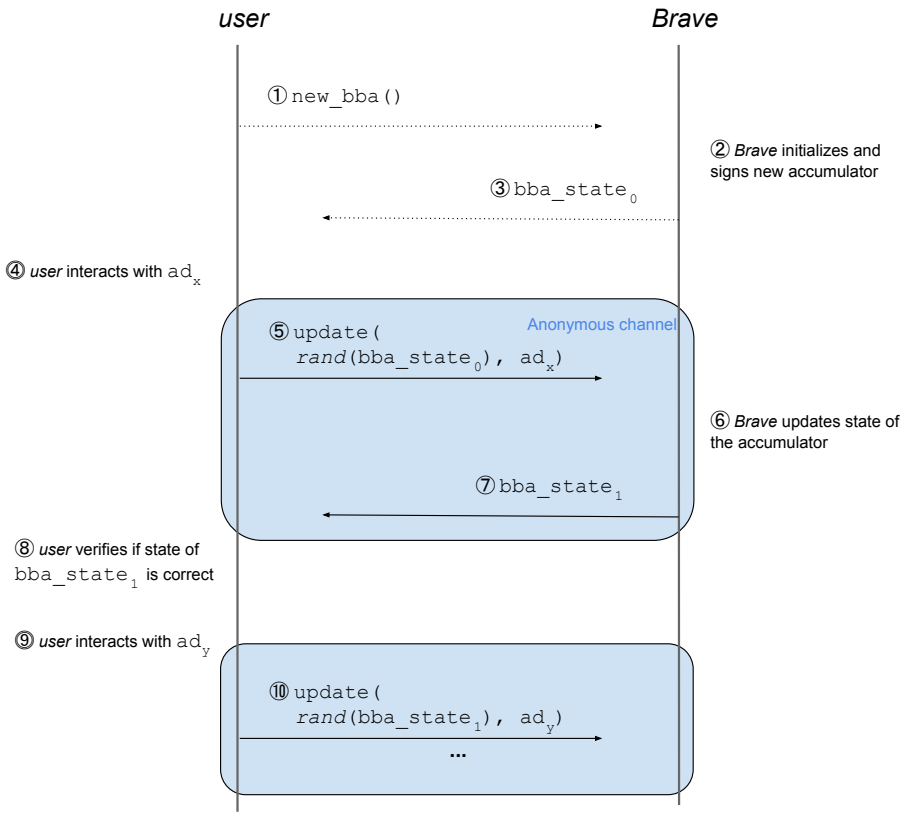

Figure 2: Interaction diagram of how *user* and *Brave* interact to initialize and update the BBA encoding the state of the rewards. Dashed lines represent a public channel; solid lines represents an anonymous channel (i.e., no linkability between the request and requester).

④ The *user* interacts with different ads through the browser;

⑤ The *user* makes a state update request, for which it randomizes the BBA and sends it to the server, together with a notification of the ads viewed; this request must be done through an *anonymous channel*, in order to preserve Properties 1 and 2;

⑥ *Brave* applies the state update to the commitment and signs it, returning the updated BBA to the *user*;

⑦ *Brave* sends the updated BBA back to the user;

⑧ The user then verifies if the new BBA state corresponds to what is expected, based on the update requests of step ⑤.

The underlying aim of BBAs is to provide the following guarantees, as introduced in Section 2.2, in the context of THEMISv2: Property 1 (privacy-preserving ad interaction); Property 2 (ad interaction unlinkability); and Property 4 (client verifiability).

3.3 Reward Calculation and Verification Protocol

The reward calculation phase consists of calculating the BAT rewards that a *user* should receive, based on the state of their BBA. The reward calculation is performed by the *user* locally. Thus, in order to prove the validity of the reward calculation (Property 5 — reward verifiability), the *user* must generate a proof of correct computation of the reward, based on the final state of the BBA and as well as how much BAT each ad should pay.

In THEMISv2, the reward calculation logic is identical to that of the initial THEMIS proposal [2]. In the initial protocol proposal, the reward was calculated by performing the scalar product between the vector with ad interactions and a vector encoding how much each ad should pay. We call the vector, which describes how much BAT should be paid by each ad interaction `policy_vector`. For example, if the current ad interactions vector was:

$$\texttt{int\_state} = [0, 0, 1, 0, 4], \tag{1}$$

and

$$\texttt{policy\_vector} = [2, 3, 3, 3, 5], \tag{2}$$

then the reward the *user* should receive is calculated using the scalar product operation over both vectors:

$$\begin{aligned} &\texttt{reward} = \langle \texttt{int\_state}, \texttt{policy\_vector} \rangle = \\ &0 \times 2 + 0 \times 3 + 1 \times 3 + 0 \times 3 + 4 \times 5 = 23 \end{aligned} \tag{3}$$

In the THEMISv2 protocol, the BBA *contains* the ad interaction vector, as it encodes the interactions with ads by the user over time. The *user* calculates locally the amount of BAT to be paid and, through a zero knowledge scheme, proves that the reward is correctly calculated according to their BBA state and the `policy_vector`. The reward request can be verified by every participant, without leaking which ads the *user* has interacted with over time.

In addition to proving that the reward is the result of the scalar vector between the `policy_vector` and the state encoded in the BBA, the *user* needs to prove that the BBA used in the rewards calculation is valid, with respect to *Brave*'s public key. Thus, any participant can trust that the BBA used to calculate the reward has been updated by *Brave*.

THEMISv2 has no constraints as to what type of Zero-Knowledge Proofs to utilize. In the technical document describing BBAs, we give details on how the reward request is computed, and the exact relation being proved by the user. This proof can be verified by *Brave*, making sure that a *user* is not being payed more than the interactions it has notified. However, this does not

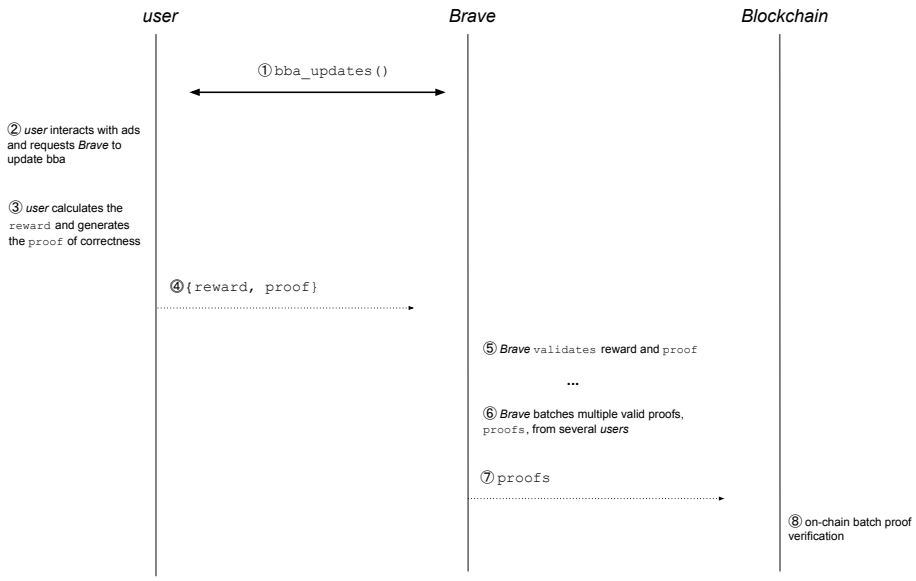

Figure 3: Interaction for off-chain and public reward calculation verification scheme for THEMISv2.

provide public verifiability of the reward computation. To provide the latter, these proofs need to be validated by a public blockchain. However, scaling the number of on-chain proof verifications to the number of Brave users is not trivial. Hence, *Brave* (or any other entity) needs to batch the proofs of reward computation before posting them on-chain, thus allowing the system to scale. This technique can be compared to ZK-Rollups [5], where a single proof validates multiple transactions. The scalability challenge of this RFC&C is to maximize the number of reward requests that can be verified on-chain per month, while optimizing the transaction costs.

Figure 3 outlines the interaction flow between the *user* and *Brave* at reward request time. The proofs generated by *users* can be verified by any participant who knows the public key of *Brave*.

① Similarly to what was described in Figure 2, the *user* interacts with *Brave* to issue a new BBA;

② The *user* interacts with *Brave* to update the BBA;

③ After multiple ad interactions, the *user* calculates the reward based on the BBA state and generates the proofs of correctness;

④ The *user* exposes the reward computation and the proof of correct computation to *Brave*;

⑤ *Brave* verifies whether the reward and proof are valid; if that is the case, *Brave* can safely pay out the reward;

⑥ *Brave* batches multiple valid proofs from several users;

⑦ *Brave* submits the batched proof to a public blockchain that supports the batched proof verification;

The reward calculation verification scheme aims to provide the following guarantees in the context of THEMISv2 (Section 2.2): Property 1 (privacy preserving ad interaction); Property 2 (ad interaction unlinkability); and Property 5 (reward verifiability).

3.4 Transparent Ad Attribution System (TAA)

The goal of the Transparent Ad Attribution system (TAA) is to provide *advertisers* with ad campaign performance metrics (i.e., metrics on how effectively their campaigns are running, engagement, budget expenditure, etc.), without the need to trust Brave. The intuition of the TAA is the following: if an *advertiser* has access to the flow of reward update requests from the *users*, the *advertiser* can have knowledge of the budget *upper bound* that the campaign is consuming over time and trust the associated ad campaign metrics.

In the context of THEMISv2, the TAA implements a many-to-many broadcast channel between the *users* (source), Brave, and the *advertisers* (destinations) (Figure 4). Every reward update request made by the *users* is encrypted and published in the broadcast channel, over which Brave and all *advertisers* have read access. Since we want to guarantee that each *advertiser* can read only messages from their own campaign, we need to enforce an access control policy to the encrypted messages published in the broadcast channel. To that end, the TAA leverages broadcast encryption [6], to ensure that each advertiser can only decrypt the messages that correspond to their campaign. Brave can decrypt all messages published in the broadcast channel in order to process them and update the interaction state of the user.

Based on Section 3.2, the reward update request contains information about i) which ad the *user* interacted with, and ii) the type of interaction. Thus, by giving *advertisers* access to the stream of requests associated with their ad campaigns – the same stream that Brave uses to update the state of the *users* – we make sure that *advertisers* have a complete and accurate view of how their campaign budgets are being spent.

The TAA aims at providing the following guarantees in the context of THEMISv2 (Section 2.2): Property 3 (privacy-preserving ad interactions) and Property 6 (advertiser verifiability).

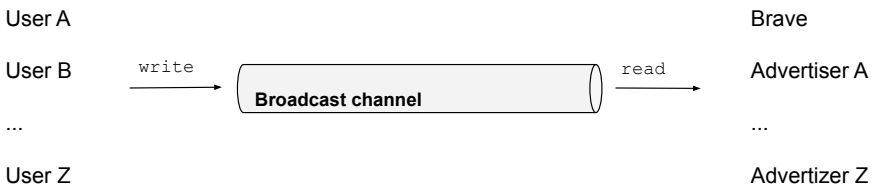

Figure 4: TAA as a broadcast channel for reward updates between *users*, *advertisers* and Brave.

4 RFC&C Community Event

The Request For Comments and Code (RFC&C) community event is a three-month event where the Brave Research team, researchers, crypto companies, and project will engage in technical discussions and collaboration to push THEMISv2 forward. The RFC&C is a great opportunity for crypto projects to actively shape the future of the BAT ecosystem.

The RFC&C consists of three distinct phases.

Phase I (until *mid-February*) Brave Research team will open a Discord channel to the community and organize public sessions where the THEMISv2 components are presented and discussed. The goal of the first phase is to introduce THEMISv2 to the community, and make sure the teams who want to participate in the RFC&C event are prepared to contribute.

Phase II (∼1 month duration, starting in *mid-February*) In the second phase, the participants will work on their submissions. A submission should focus on one (or more) of the the following paths:

1. To propose, build, and measure zero-knowledge schemes for scaling THEMISv2;
2. To suggest changes to the THEMISv2 protocol; or
3. To propose and implement a preliminary evaluation, of a completely new protocol.

We plan to consider and review any reasonable suggested modification to the protocol regardless of how much they change the current design, as described in this document (e.g., replacing the SNARKs-based verification on-chain by a scalable L1 option or optimistic - rollups). However, we expect the goals and requirements described in this document to hold true as a result of the proposed modifications. The Brave Research team plans to run individual sessions with participants, where we can discuss the protocol details and open the floor to any questions. In addition, there will be a public Discord channel to answer questions the participants may have during this phase.

Phase III (∼2 weeks duration, starting in *mid-March*) The third phase of the event consists of the Brave Research team reviewing the submissions and reaching out for potential partnerships. The reviewing process will consider all comments and technical suggestions. The submissions may range from proposing changes to our initial protocol, to a completely new design and MVP (e.g., the reward calculation being performed by a L1 blockchain, etc).

4.1 Submissions

In addition to the protocol requirements outlined in Section 2.2, the following should be met through the preliminary evaluation:

Costs: Price of on-chain reward verification of below $0.1 (per reward);

Scalability 1: The target protocol is able to support up to 5M concurrent users;

Scalability 2: The target protocol is able to verify 50M rewards per month and to prove their validity on-chain;

Client-side performance: Many clients will run on mobile devices, hence the computation and communication requirements must be kept low.

Our current plan is to evaluate the submissions based on a combination of the above, together with the level of decentralisation and the complexity of maintenance in production.

4.2 Open Challenges and Directions to Explore

Below we present a list of some of the open challenges of THEMISv2 as well as some possible directions to explore. The solutions to these might defer from team to team. In Section 4.1 we explain how we evaluate the different submissions.

- What is the best proof system for THEMISv2. This question has many dimensions to it:
  - What are the requirement for a trusted setup?
  - What are the computational complexity and resources required to generate the proof on the client-side, as the proof will, ultimately, be generated on the *user*'s device? Brave browser is very conscious about battery consumption [7, 8]);
  - How does the proof artefacts affect the costs and scalability of the on-chain proof verification?
  - What is the upper bound for the resources required to verify and batch the proofs from multiple users (we expect, in the first phase to perform about 10M reward verifications per month?

  - Which tools and frameworks can we use to implement and test the proof circuits?

- Are there services that could perform the proof verification and batching for Brave?
- How to prove that the reward was calculated based on a BBA, state which was the result of an *opened* BBA that has been issued and updated by the expected *issuer*? We see different paths to answering this question, namely:
  1. Build a circuit that verifies the validity of the BBA and its signature and that performs the *open* within the circuit.
  2. Prove the validity of the signature *outside* independently of the circuit. This would require a separate batching for the signature verification step.
- Is the BBA construction we selected the best option for every scenario? The SPS-EQ based BBA construction seems ideal in terms of user computation and communication complexity, but is there a scenario where a ZKP-based BBA could be a better fit?
- Is Ethereum the best blockchain for verifying the batches of proofs from the reward calculation? How would it compare with other L1 blockchains in terms of scalability, costs, and degree of decentralization?
- Instead of offloading the reward calculation to the *user*, could a L1 blockchain calculate the rewards of the *users* based on the BBA state, while guaranteeing the privacy and trust requirements required by THEMISv2 2.2? If so,
  1. what are the trade-offs between the L1 and L2 approaches, considering privacy, scalability, costs and degree of decentralization?
- Could we replace the TAA (Section 3.4) with a cryptographic protocol which guarantees all requirements outlined in Section 3, namely the Properties 1, 3, and 6? If so, which one?

We encourage the teams to share their answers to some of these questions in their reports.

5 Related Work

The current advertising ecosystem abounds with issues associated with its performance, its transparency, the user's privacy and the integrity of billing and reporting. These failures are already well studied and there are numerous works aiming to shed light on how digital advertising works [9, 10, 11, 12, 13, 14].

Apart from the studies highlighting the failures of current ad delivery protocols there are also important novel ad systems proposed. In [15], Juels is the first to study private targeted advertising. Author proposes a privacy-preserving targeted ad delivery scheme based on PIR and Mixnets. In this scheme, advertisers choose a negotiant function that assigns the most fitting ads in their database for each type of profile. The proposed scheme relies on heavy cryptographic operations and therefore it suffers from intensive computation cost. Their approach focuses on the private distribution of ads and does not take into account other aspects such as view/click reporting.

In [16], authors propose Adnostic: an architecture to enable users to retrieve ads on the fly. Adnostic prefetches n ads before the user starts browsing and stores them locally. Aside from the performance benefits of this strategy, Adnostic does this prefetching also in order to preserve the privacy of the user. The parameter n is configurable: larger n means better ad matching, when smaller n means less overhead. In order for the ad-network to correctly charge the corresponding advertisers, Adnostic performs secure billing by using homomorphic encryption and zero-knowledge proofs.

In [17, 18, 19], authors propose Privad: an online ad system that aims to be faster and more private than today's ad schema. Privad introduces an additional entity called Dealer. The Dealer is responsible for anonymizing the client so as to prevent the ad-network from identifying the client and also handle the billing. To prevent the Dealer from accessing user's behavioral profile and activity it encrypts the communications between the client and the Dealer. A limitation of Privad is that Dealer is a centralized entity that needs to be always online.

In [20], authors propose ObliviAd: a provably secure and practical online behavioral advertising architecture that relies on a secure remote co-processor (SC) and Oblivious RAM (ORAM) to provide the so called secure hardware-based PIR. In ObliviAd, to fetch an ad, a user first sends their encrypted behavioral profile to the SC which securely selects the ads that match best based on the algorithm specified by the ad network. To prevent the ad-network from learning which ads are selected, they leverage an ORAM scheme. The selected ads are finally sent to the user encrypted, along with fresh tokens used to billing. User will send back one of these tokens as soon as they view/click on an ad.

In [21] authors point out that, in current advertising systems the ad-network exclusively determines the payment to get from advertisers and the revenue to share with publishers. This means that (i) a malicious ad-network can overcharge advertisers or underpay publishers. To make matters worse, as bills cannot be justified by the ad-network, malicious advertisers can deny actual views/clicks to ask for refund. On the other hand, (ii) malicious publishers may claim clicks that did not happen, in order to demand higher revenues. To address this problem of unfairness, authors propose a protocol where the ad click reports are encrypted by the user using the public key of the ad-network and signed by both publishers and advertisers.

In [22], authors use an additively encryption scheme to design a protocol that enables privacy-preserving advertising reporting at scale, without needing any trusted hardware. Performance evaluation results show that their proto-

col reduces the overhead of reporting by orders of magnitude compared to the ElGamal-based solution of Adnostic [16] (i.e., 1 MB of bandwidth per impression when handing 32,000 advertisements). Contrary to our approach, authors assume a Trusted Third Party (TTP) that owns the key for the homomorphic encryption.

In [23], authors propose CAMEO: a framework for mobile advertising that employs intelligent and proactive prefetching of advertisements. CAMEO uses context prediction, to significantly reduce the bandwidth and energy overheads, and provides a negotiation protocol that empowers applications to subsidize their data traffic costs by "bartering" their advertisement rights for access bandwidth from mobile ISPs. In [24], authors propose a location-aware, personalised and private advertising system for mobile platforms. In this system, ads are locally broadcast to users within mobile cells. The ad matching happens locally based on the user interests. Finally ad view and click reports are collected using a DTN system. In [25], authors propose a new ad protocol that uses homomorphic and searchable encryption to allow users transmit mobile sensor data to a cloud service that responds back with the best matching contextual advertisements.

In [26], authors present VEX, a protocol for ad exchanges to run low-latency and high-frequency ad auctions that are verifiable and auditable, in order to prevent fraud in a context where parties participating in the auction – bidders and ad exchanges – may not know each other. Based on their evaluation of the system, the authors claim that the additional storage required and latency imposed by VEX are low and practical in the context of ad auctions. In [27], authors present and implement PROTA, a privacy-preserving protocol for real-time advertising which uses keywords to match users interests with ads. By using bloom filters, the authors make the ad matching task efficient. The protocol relies on a trusted third party to cooperate with the ad exchange during the bidding and ad delivering phase. The authors implement and evaluate the protocol, and conclude that the time upper bond for matching ads is 200ms, which is considerable practical in the context of an ad matching system.

In [28], authors present and evaluate a system that aims at providing high-quality ad targeting in multiple scenarios, while giving the user the ability to control their privacy. The system consists of tailored extensions that *mine* the user behaviour locally with low overhead. The extensions generate user behavioural data that can be shared with advertisers without leaking undesirable user information. Similarly to THEMISv2, the authors discuss how the system can be used by users and advertisers, and how it can be used as a replacement for the tracking-based business model in the online advertising industry.

In [29], authors set out to formalize the concept of privacy in the context of the online advertising ecosystem and to develop a provably secure privacy-preserving protocol for the online advertising ecosystem. While the authors claim that the definition of privacy presented in the paper is more useful compared to previous work in the online advertising context, their attempts to develop a provably secure privacy-preserving protocol has failed due to being hard to balance privacy with usefulness of the user data. The authors conjecture

that cryptographic mechanisms have the potential to solve the privacy versus data usefulness conundrum. Using applying cryptography is the basis of how THEMISv2 proposes to preserve privacy when calculating ad rewards, providing advertisers with campaign metrics and performing confidential payments to users.

Towards a similar direction with the user rewarding schema of THEMISv2, in [30], authors propose a privacy-aware framework to promote targeted advertising. In this framework, an ad broker responsible for handling ad targeting, sits between advertisers and users and provides certain amount of compensation to incentivize users to click ads that are interesting yet sensitive to them. In [31], authors propose a targeted advertising framework which enables users to get compensated based on the amount of user tracking they sustain and the privacy they lose. The authors analyze the interaction between the different parties in the online advertising context — advertisers, the ad broker and users — and propose a framework where the interactions between the different parties are a positive-sum game. In this game, all parties are incentivized to behave according to what other parties expect, achieving an equilibrium where everyone benefits. More specifically, the users determine their click behaviour based on their interested and their privacy leakage, which in turn will influence the advertisers and ad broker to provide less invasive and better ads. THEMISv2 relies on a similar game theoretical approach. By providing compensation for good behaviour while providing the verification mechanisms for all parties to audit whether everyone is behaving according to the protocol, the incentives to cheat and misbehave are lower.

References

[1] Brave Software Inc. Brave - BAT - Whitepaper. https://basicattentiontoken.org/BasicAttentionTokenWhitePaper-4.pdf, 2017.

[2] Gonçalo Pestana, Iñigo Querejeta-Azurmendi, Panagiotis Papadopoulos, and Benjamin Livshits. Themis: Decentralized and trustless ad platform with reporting integrity, 2020.

[3] Tibor Jager and Andy Rupp. Black-box accumulation: Collecting incentives in a privacy-preserving way. *Proc. Priv. Enhancing Technol.*, 2016(3):62–82, 2016.

[4] Jan Bobolz, Fabian Eidens, Stephan Krenn, Daniel Slamanig, and Christoph Striecks. Privacy-preserving incentive systems with highly efficient point-collection. In Hung-Min Sun, Shiuh-Pyng Shieh, Guofei Gu, and Giuseppe Ateniese, editors, *ASIA CCS '20: The 15th ACM Asia Conference on Computer and Communications Security, Taipei, Taiwan, October 5-9, 2020*, pages 319–333. ACM, 2020.

[5] ethhub.io. ZK-Rollups. https://docs.ethhub.io/ethereum-roadmap/layer-2-scaling/zk-rollups/, 2020.

[6] Amos Fiat and Moni Naor. Broadcast encryption. In *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, volume 773 of *Lecture Notes in Computer Science*, pages 480–491. Springer, 1993.

[7] Matteo Varvello, Kleomenis Katevas, Mihai Plesa, Hamed Haddadi, and Benjamin Livshits. Batterylab, a distributed power monitoring platform for mobile devices. *Proceedings of the 18th ACM Workshop on Hot Topics in Networks*, Nov 2019.

[8] Mohammad Ghasemisharif, Peter Snyder, Andrius Aucinas, and Benjamin Livshits. Speedreader: Reader mode made fast and private, 2018.

[9] Daniel G Goldstein, R Preston McAfee, and Siddharth Suri. The cost of annoying ads. In *Proceedings of the 22nd international conference on World Wide Web*, 2013.

[10] Narseo Vallina-Rodriguez, Jay Shah, Alessandro Finamore, Yan Grunenberger, Konstantina Papagiannaki, Hamed Haddadi, and Jon Crowcroft. Breaking for commercials: characterizing mobile advertising. In *Proceedings of the Internet Measurement Conference*, 2012.

[11] Michalis Pachilakis, Panagiotis Papadopoulos, Evangelos P Markatos, and Nicolas Kourtellis. No more chasing waterfalls: A measurement study of the header bidding ad-ecosystem. In *Proceedings of the 19th Internet Measurement Conference*, 2019.

[12] Phillipa Gill, Vijay Erramilli, Augustin Chaintreau, Balachander Krishnamurthy, Konstantina Papagiannaki, and Pablo Rodriguez. Best paper–follow the money: understanding economics of online aggregation and advertising. In *Proceedings of the Conference on Internet measurement conference*, 2013.

[13] Alexey Reznichenko, Saikat Guha, and Paul Francis. Auctions in do-not-track compliant internet advertising. In *Proceedings of the 18th ACM conference on Computer and communications security*, 2011.

[14] Panagiotis Papadopoulos, Nicolas Kourtellis, Pablo Rodriguez Rodriguez, and Nikolaos Laoutaris. If you are not paying for it, you are the product: How much do advertisers pay to reach you? In *Proceedings of the Internet Measurement Conference*, 2017.

[15] Ari Juels. Targeted advertising ... and privacy too. In *Proceedings of the Conference on Topics in Cryptology: The Cryptographer's Track at RSA*, 2001.

[16] Vincent Toubiana, Arvind Narayanan, Dan Boneh, Helen Nissenbaum, and Solon Barocas. Adnostic: Privacy preserving targeted advertising. In *Proceedings Network and Distributed System Symposium*, 2010.

[17] Hamed Haddadi, Saikat Guha, and Paul Francis. Not all adware is badware: Towards privacy-aware advertising. In *Conference on e-Business, e-Services and e-Society*, 2009.

[18] Saikat Guha, Alexey Reznichenko, Kevin Tang, Hamed Haddadi, and Paul Francis. Serving ads from localhost for performance, privacy, and profit. In *HotNets*, pages 1–6, 2009.

[19] Alexey Reznichenko and Paul Francis. Private-by-design advertising meets the real world. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2014.

[20] Michael Backes, Aniket Kate, Matteo Maffei, and Kim Pecina. Obliviad: Provably secure and practical online behavioral advertising. In *IEEE Symposium on Security and Privacy*, 2012.

[21] J. Hua, A. Tang, and S. Zhong. Advertiser and publisher-centric privacy aware online behavioral advertising. In *2015 IEEE 35th International Conference on Distributed Computing Systems*, June 2015.

[22] Matthew Green, Watson Ladd, and Ian Miers. A protocol for privately reporting ad impressions at scale. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[23] Azeem J. Khan, Kasthuri Jayarajah, Dongsu Han, Archan Misra, Rajesh Balan, and Srinivasan Seshan. Cameo: A middleware for mobile advertisement delivery. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '13, page 125–138, New York, NY, USA, 2013. Association for Computing Machinery.

[24] Hamed Haddadi, Pan Hui, and Ian Brown. Mobiad: Private and scalable mobile advertising. In *Proceedings of the Fifth ACM International Workshop on Mobility in the Evolving Internet Architecture*, 2010.

[25] Debmalya Biswas and Krishnamurthy Vidyasankar. Privacy preserving and transactional advertising for mobile services. *Computing*, 96(7):613–630, 2014.

[26] Sebastian Angel and Michael Walfish. Verifiable auctions for online ad exchanges. *SIGCOMM Comput. Commun. Rev.*, 43(4):195–206, August 2013.

[27] Yiming Pang, Bo Wang, Fan Wu, Guihai Chen, and Bo Sheng. Prota: A privacy-preserving protocol for real-time targeted advertising. In *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)*. IEEE, 2015.

[28] M. Fredrikson and B. Livshits. Repriv: Re-imagining content personalization and in-browser privacy. In *2011 IEEE Symposium on Security and Privacy*, SP'11, 2011.

[29] A. Mandal, J. Mitchell, H. Montgomery, and A. Roy. Privacy for targeted advertising. In *2017 IEEE Conference on Communications and Network Security (CNS)*, pages 438–443, 2017.

[30] Wei Wang, Linlin Yang, Yanjiao Chen, and Qian Zhang. A privacy-aware framework for targeted advertising. *Computer Networks*, 79:17–29, 2015.

[31] Javier Parra-Arnau. Pay-per-tracking: A collaborative masking model for web browsing. *Information Sciences*, 385:96–124, 2017.