Verifiable Auctions for Online Ad Exchanges

Sebastian Angel and Michael Walfish
The University of Texas at Austin
Ad Exchanges allow real time selling and buying of ad space
Ad Exchanges allow real time selling and buying of ad space

d this ad get here?
Publisher

20-30 year old, male, Texas resident

Ad Exchange
20-30 year old, male, Texas resident
telegraph.co.uk, ad unit 5, 300x250
20-30 year old, male, Texas resident
telegraph.co.uk, ad unit 5, 300x250
20 year old, male, Texas resident

Ad Exchange

$10

Zappos

$5

Blue Cross Blue Shield

$20

GEICO

Tuesday, August 13, 13
Second-price auction
Winner = highest bidder
Price = second-highest bid

Winner
$10

$5

$20

Ad Exchange

Tuesday, August 13, 13
The Telegraph

The Ashes: Flower: Broad had every right not to walk • 'Australia lose to DR's

Charity donations 'go missing' as watchdog investigates website

Gavin Fuller sets a quiz to find out just how much you really love the UK.

JK Rowling's crime novel: 'work of master storyteller'

To middle-class Leftists, workers are an exotic tribe

pictures: follow Telegraph Fashion's style guide for men on x to wear shorts in the city with ease.

Comment & Blogs » Search - enhanced by Google

Tuesday, August 13, 13
Today’s ecosystem is unnecessarily predicated on trusting ad exchanges, disenfranchising advertisers and publishers.

Our goal is to remove that trust ...

... by making auctions verifiable.
Respecting today’s ecosystem is technically challenging

- Need to support millions of auctions per second
- Need to add little latency to Web page loads
- Cannot disclose submitted bids
- Cannot require participants to know each other
- Cannot introduce trusted third parties (DNS is okay)
More motivation

Verifiability in ad exchanges would:

• Deter potential misbehavior
• Strengthen the service provided by ad exchanges
• Democratize the auctioneer function
1. What are some issues with ad exchanges?

2. How do we provide verifiability?

3. What is the cost of verifiability?
The status quo has several weaknesses

Publisher → Ad Exchange

$10

$5

$10

$20
The status quo has several weaknesses
The status quo has several weaknesses
The status quo has several weaknesses

Publisher

$6

Ad Exchange

Receives $10 from Advertiser but reports only $6

$10

$5

$20

Zappos.com

BlueCross BlueShield

GEICO
Issues not addressed in this talk

• Publisher may not deliver ads to end users

• An ad exchange may refuse to run auctions
1. What are some issues with ad exchanges?

2. How do we provide verifiability?

3. What is the cost of verifiability?
Notable aspects of VEX's design

- It contains two phases
  - Auction
  - Audit (offline)
Status Quo

ADX

Bid

Auction

Outcome

VEX

ADX

Encoded Bid

List of Encoded Bids

Bid

Verify auction
Notable aspects of VEX’s design

• It contains two phases
  • Auction
  • Audit (offline)

• It introduces a technique for order comparisons against hidden integers
Notable aspects of VEX's design

- It contains two phases
  - Auction
  - Audit (offline)

- It introduces a technique for order comparisons against hidden integers

- It requires some engineering
Notable aspects of VEX's design

- It contains two phases
- It introduces a technique for order comparisons against hidden integers
- It requires some engineering

Auction
Audit (offline)
The Telegraph | ADX | Zappos

--- Request --- | --- Request --- | --- Encoded Bid generation ---

\langle b_i \rangle
(1) Ensure that the same set of encoded bids is received by all participants

(2) Ensure that the set of encoded bids includes all advertisers’ submissions
Encoded Bid generation
Sharing of Encoded Bids
Disclosure
Auction computation

The Telegraph
Request → Request

ADX
← ⟨b₁⟩, ..., ⟨bₙ⟩ ← ⟨b₁⟩, ..., ⟨bₙ⟩

Zappos

Is the outcome correct?
Structure of a correct second-price auction

Inputs: Bids = \{b_1, ..., b_N\}
Output: winner \(w\), price \(p\)

Correctness

1. There exists a bid \(b_i = p\)
2. There are \(N-2\) bids \(\leq p\)
3. There exists a bid \(b_w \geq p\)

\(b_w\) = highest bid
\(p\) = second-highest bid
Structure of a correct second-price auction

Inputs: Bids = \{b_1, ..., b_N\}

Output: winner \( w \), price \( p \)

Correctness

1. There exists a bid \( b_i = p \)
2. There are \( N-2 \) bids \( \leq p \)
3. There exists a bid \( b_w \geq p \)

How can the verifier check that the correctness conditions are satisfied?
**Diagram Description:**

- **The Telegraph** sends a request to **ADX**.
- **ADX** encodes the bids and sends them back to **The Telegraph**.
- **The Telegraph** then sends the encoded bids to **Zappos**.
- **Zappos** shares the encoded bids and disclosures with **ADX**.
- **ADX** computes the auction outcome.
- **The Telegraph** receives the outcome.
- **Proof generation** is initiated by **ADX**.
- **Proofs please** are sent to **The Telegraph**.
- **The Telegraph** receives the proofs and verifies the auction.

**Notes:**
- The diagram includes a reference to **Proofs please** and **Proofs**.
- There is a date reference in the bottom left corner: "Tuesday, August 13, 13."
A bid encoding is the last node of a hash chain, where the length of the chain is related to the bid.

Notation: $H$ is a cryptographic hash function

inputs: $\text{bid} = 4$, $\text{seed} = \{0,1\}^n$

Encoded bid $\langle \text{bid} \rangle = H^{\text{bid} + 1}(\text{seed}) = H^5(\text{seed}) = H(H(H(H(H(\text{seed}))))))$
A bid encoding is the last node of a hash chain, where the length of the chain is related to the bid.

Notation: $H$ is a cryptographic hash function.

Inputs: $\text{bid} = 4$, $\text{seed} = \{0,1\}^n$

Encoded bid $\langle \text{bid} \rangle = H^{\text{bid}+1}(\text{seed}) = H^5(\text{seed}) = H(H(H(H(H(\text{seed}))))))$

Given $\langle \text{bid} \rangle$, it is hard to determine # nodes (i.e., the bid + 1).
A bid encoding is the last node of a hash chain, where the length of the chain is related to the bid.

Notation: \( H \) is a cryptographic hash function.

Given \( \langle \text{bid} \rangle \), it is hard to determine # nodes (i.e., the bid + 1).

How can the party generating this encoded bid prove that the secret bid is \( \geq \) some integer?

A bid encoding is the last node of a hash chain, where the length of the chain is related to the bid.

Given \( \langle \text{bid} \rangle \), it is hard to determine # nodes (i.e., the bid + 1).
A bid encoding is the last node of a hash chain, where the length of the chain is related to the bid.

Notation: $H$ is a cryptographic hash function

Given $\langle \text{bid} \rangle$, it is hard to determine # nodes (i.e., the bid + 1)

How can the party generating this encoded bid prove that the secret bid is $\geq$ some integer?

By providing an earlier node in the chain as proof.

Request $\langle b_i \rangle$ ➔ Encoded Bid generation ➞

Inputs:
- bid = 4
- seed = {0,1}

Seed

Origin

$H \xrightarrow{} \text{H} \xrightarrow{} \text{H} \xrightarrow{} \text{H} \xrightarrow{} \text{H} \xrightarrow{} \text{H} \xrightarrow{} \langle 4 \rangle$

Encoded Bid generation
A bid encoding is the last node of a hash chain, where the length of the chain is related to the bid.

Notation: \( H \) is a cryptographic hash function.

Inputs: \( \text{bid} = 4, \ seed = \{0,1\}^n \)

Encoded bid \( \langle \text{bid}\rangle = H^{\text{bid}+1}(\text{seed}) = H^5(\text{seed}) = H(H(H(H(H(\text{seed}))))) \)

Given \( \langle \text{bid}\rangle \), it is hard to determine \# nodes (i.e., the bid + 1).
How to prove equality (==)?

How to prove less-than-or-equal-to (≤)?
How to prove equality (==)?

• Can be proven if the seed is distinguishable from a node in the chain

How to prove less-than-or-equal-to (≤)?
How to prove equality (==)?

• Can be proven if the seed is distinguishable from a node in the chain

How to prove less-than-or-equal-to (≤)?

• Requires an application-specific maximum value M to be set ahead of time

• Encoded bids are of the form $M - x$, where $x$ is the desired value

• Proving that $x$ is $\leq y$ from an encoded bid is done by encoding the value $M - x$, and proving that $M - x \geq M - y$ (which implies $x \leq y$)
Encrypted Bids generation

Sharing of Encoded Bids

Auction computation

Proof generation

Proofs please

Proofs

Auction verification

Demand

Supply

Outcome
Limitations

• Ad Exchange can use historical knowledge to introduce fake bids

• Right of first refusal given to a colluding bidder

• Sale price is disclosed

• Comparison scheme performs well only for small ranges
1. What are some issues with ad exchanges?

2. How do we provide verifiability?

3. What is the cost of verifiability?
Evaluation questions

(1) How much latency is introduced by generating and sharing the encoded bids?

(2) How many auctions can the ad exchange handle under this verifiable regime?

(3) What is the cost of auditing?
Implementation

- Baseline (status quo) is ~1800 lines of C++
Implementation

- Baseline (status quo) is ~1800 lines of C++
- VEX requires ~2000 additional lines of C++
Implementation

- Baseline (status quo) is ~1800 lines of C++
- VEX requires ~2000 additional lines of C++
- SHA-256 is used to generate the encodings
- We use the ESIGN signature scheme (signatures are required by the protocol for sharing the encoded bids)
- Bids range from 1 ($0.01) to M = 10,000 ($100)
Evaluation method and testbed setup

All experiments were run on Emulab:

Each machine has four 2.2 Ghz 64-bit 8-core processors, 128 GB RAM, Ubuntu 12.04

1Gbps links, fixed 10 ms latency, zero packet loss
(1) How much latency is introduced by generating and sharing the encoded bids?

- **The Telegraph**
  - Request
  - $\langle b_1, \ldots, b_N \rangle \leftarrow$
  - Outcome

- **ADX**
  - Request
  - $\langle b_i \rangle \leftarrow$
  - Outcome

- **Zappos**
  - Commitment generation
  - Sharing of commitments
  - Disclosure
  - Auction computation
Most of VEX’s latency overhead comes from an extra round of communication

# of advertisers: 20
Most of VEX’s latency overhead comes from an extra round of communication

- ADX: < 1 ms
- Advertiser: 0.5 - 2.5 ms
- Publisher: < 1 ms

# of advertisers: 20
(2) How many auctions can the ad exchange handle under a verifiable regime?
VEX has modest overhead that decreases as the number of bidders increases.
(3) What is the cost of auditing?
The costs of verifying an auction scales linearly with the number of bidders.
The costs of verifying an auction scales linearly with the number of bidders.

Ad Exchange’s proof generation

Auction-to-audit ratio: 150
Storing encoded bids for on-demand auditing is not completely unreasonable

Cost of storing all needed metadata for one week’s worth of auctions (25.2 billion auctions, 50 advertisers/auction) on Amazon S3: $43,000 / month
Summary of evaluation results

• VEX introduces ~50 ms of latency to every auction (most of which comes from an additional round of communication)

• For a 20 bidder auction, VEX’s ad exchange can handle 1/2 the number of auctions of a baseline (unverifiable) protocol

• VEX’s audits are costly (150 times more expensive than running auctions), but the needed metadata is small enough and can be stored for later auditing
Related Work

Verifiable auctions

Straight line computation (SLC) [Rabin et al. ICALP’12]
Distributed auctioneer [Lipmaa et al. CRYPTO’02]

Online Advertising

User Privacy [Privad NSDI11, Adnostic NDSS’10]
Fraud [Stone-Gross et al. IMC’11, Dave et al. SIGCOMM’12]
Ad Exchanges [Muthukrishnan, WINE’09]

Privacy-preserving integer comparisons

Summary and future work

• VEX provides verifiability at modest costs

• VEX relies on an efficient mechanism that allows privacy-preserving integer comparisons, coupled with engineering

• Future directions:
  • Providing verifiability to other advertising payment models (pay-per-click, pay-per-action)
  • Verifying the delivery of ads to end-users