A Hash-Cash Based Music Streaming Payment System

Timothy Chen
San Jose State University
Fall 2014
AGENDA

- Introduction
- Background
- Features implemented
- Issues Encountered During Testing
- Conclusion
INTRODUCTION

- There are many popular music streaming services
  - Pandora
  - Spotify

- Many of these pay using a royalty system
  - Artist paid by number of streams

- My project is to create a new payment service based on a crypto currency like set-up
MY PROJECT DESIGN

I built

- A web front end
  - Upload music in MP3 format
  - Play music
  - A ranking tool
  - Verify crypto-coin tool

- A back end
  - Uses artist’s name, content of music, timestamp, music listener’s IP address as seed in a hash cash SHA256 function for artist to earn the new crypto-currency
  - Mining process is run in parallel while music is playing
BACKGROUND

- **ASCAP**
  - American Society of Composers, Authors, and Publishers
  - Protects its members’ rights by licensing, distributing royalties, and copyright for the music publicly

- **Payola**
  - The illegal paying of cash or gifts in exchange for airplay

- **Streaming music websites**
  - Pandora
  - Spotify

- **Bitcoin**

- **SHA256**
BACKGROUND

- Distribution of Royalty System for Spotify

1. SPOTIFY MONTHLY REVENUE

2. ARTIST’S SPOTIFY STREAMS / TOTAL SPOTIFY STREAMS

3. ~70% TO MASTER & PUBLISHING OWNERS

4. ARTIST’S ROYALTY RATE

5. ARTIST PAYOUT
## Royalty System Issues

**Music Streaming Price Index as of Feb 1, 2014**

<table>
<thead>
<tr>
<th>Store</th>
<th>Per Stream</th>
<th>Per Song Download</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nokia</td>
<td>0.07411</td>
<td>9</td>
</tr>
<tr>
<td>Google Play</td>
<td>0.04573</td>
<td>15</td>
</tr>
<tr>
<td>Xbox Music</td>
<td>0.03212</td>
<td>22</td>
</tr>
<tr>
<td>simfy</td>
<td>0.01626</td>
<td>43</td>
</tr>
<tr>
<td>Napster</td>
<td>0.01578</td>
<td>44</td>
</tr>
<tr>
<td>MediaNet</td>
<td>0.01140</td>
<td>61</td>
</tr>
<tr>
<td>Rhapsody</td>
<td>0.01122</td>
<td>62</td>
</tr>
<tr>
<td>Muve Music</td>
<td>0.00875</td>
<td>80</td>
</tr>
<tr>
<td>Deezer</td>
<td>0.00754</td>
<td>93</td>
</tr>
<tr>
<td>Rdio</td>
<td>0.00692</td>
<td>101</td>
</tr>
<tr>
<td><strong>Spotify</strong></td>
<td><strong>0.00521</strong></td>
<td><strong>134</strong></td>
</tr>
<tr>
<td>MySpace Music</td>
<td>0.00094</td>
<td>745</td>
</tr>
<tr>
<td>Amazon Cloud</td>
<td>0.00012</td>
<td>5,862</td>
</tr>
</tbody>
</table>

* Indie Label Catalog of 1,500 Songs
* Sales for Calendar Years 2012-2013
* These Streaming Rates before Dist Fee's
* Per Song Download Ratio @ .70
## My System vs Spotify’s

<table>
<thead>
<tr>
<th></th>
<th>100 user listen for 5 minutes</th>
<th>Artist earns</th>
</tr>
</thead>
<tbody>
<tr>
<td>My System</td>
<td>$1000 * (500 the artist own coins / 1000 total coins)*(my cut 10%)</td>
<td>$450</td>
</tr>
<tr>
<td>Royalty</td>
<td>$1000*(100 streams / 600 streams) * (70% website cut)</td>
<td>$50</td>
</tr>
</tbody>
</table>
**Bitcoin**

- Bitcoin uses peer-to-peer technology
- No central authority or banks
- Managing transactions and the issuing of bitcoins is carried out collectively by the network
- Bitcoin is open-source
- Its design is public
  - Nobody owns or controls Bitcoin and everyone can participate
**Bitcoin Transactions**

- **Public key**
  - Verify the signatures to check the chain of ownership

- **Private key**
  - Proves who is the owner of a bitcoin
Bitcoin Transactions

Transaction

Owner 1's Public Key

Hash

Owner 0's Signature

Sign

Owner 1's Private Key

Transaction

Owner 2's Public Key

Hash

Owner 1's Signature

Sign

Owner 2's Private Key

Transaction

Owner 3's Public Key

Hash

Owner 2's Signature

Sign

Owner 3's Private Key

Verify
**Timestamp Server**

- **Peer-to-peer**
  - Lets everybody know a bitcoin’s history

- **Prevents the use of the same bitcoin twice**
  - Keeps track of who is the owner of a bitcoin at a given time

- **All transactions have to be public**
  - Notification sent to bitcoin network when a transaction is completed
INCENTIVE

- Earn bitcoins
  - Verify others’ transactions
  - The number of leading zeros required is increasing to do verification
  - Number of people mining is increasing
MINING REQUIREMENTS

- CPU
- Graphics card
  - AMD (ATI)
  - Nvidia
- Mining Tool
  - ASIC
Bitcoin Mining
My Bitcoin Account

- I got 0.00000875 bitcoin for one day
- Nvidia GTX 670

Account Details for timchen623

Unconfirmed income is added to your balance if and when blocks are confirmed. To improve your income per block, improve your score in the shifts eligible for payments by increasing your hash power.

<table>
<thead>
<tr>
<th>Personal Assets</th>
<th>Balance</th>
<th>Unconfirmed</th>
<th>Future</th>
<th>Expected per block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bitcoins [send]</td>
<td>0.00001243</td>
<td>-</td>
<td>0.00001243</td>
<td>0.00000423</td>
</tr>
<tr>
<td>Namecoins [send]</td>
<td>0.00002894</td>
<td>-</td>
<td>0.00002894</td>
<td>0.00000846</td>
</tr>
</tbody>
</table>
MINING PERFORMANCE COMPARISON

Bitcoin Mining Performance Comparison
diagcn kernel vs. CUDA-optimized rpcminer
AMD vs. Nvidia

<table>
<thead>
<tr>
<th>Graphics Card</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMD HD 7970</td>
<td>565</td>
</tr>
<tr>
<td>AMD HD 7950</td>
<td>450</td>
</tr>
<tr>
<td>Nvidia GTX Titan</td>
<td>340</td>
</tr>
<tr>
<td>AMD HD 7790</td>
<td>272</td>
</tr>
<tr>
<td>Nvidia GTX 680</td>
<td>126</td>
</tr>
</tbody>
</table>

Source: http://www.ExtremeTech.com
# Price Comparison

**High End Videocards - Updated 11th of March 2014**

<table>
<thead>
<tr>
<th>Videocard</th>
<th>Price (USD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeForce GTX Titan</td>
<td>$999.99*</td>
</tr>
<tr>
<td>GeForce GTX 680</td>
<td>$359.99</td>
</tr>
<tr>
<td>Radeon HD 7970 / R9 280X</td>
<td>$299.99*</td>
</tr>
<tr>
<td>Radeon HD 7950</td>
<td>$359.99</td>
</tr>
</tbody>
</table>
SHA256

- Hash function computed with 32-bit words

- One-way hashing method
  - Cannot be reversed to original value

- Implement version SHA256 both in JavaScript and WebCL
FEATURES IMPLEMENTED IN THE MP3-BASED CURRENCY SYSTEM

- System to upload music
- Generate a hash string for each individual listening coin
- Use the SHA256 method to mine for listening coins
- System to save the coin after the artist receives it
Features implemented in An MP3-based Currency System

- Rank for who has the highest listening coin count for each month
- Verification check for each transaction
UPLOAD THE MUSIC FLOWCHART

1. Pass MP3 file

Web Front End → Upload Music

2. Store data

Upload Music → Database
UPLOAD THE MUSIC APPLICATION

artist Rank verification coin tool

<table>
<thead>
<tr>
<th>Music</th>
<th>Player</th>
<th>artist name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1450.mp3</td>
<td>2:37</td>
<td>Tim</td>
</tr>
<tr>
<td>2537.mp3</td>
<td>2:43</td>
<td>Tom</td>
</tr>
<tr>
<td>2555.mp3</td>
<td>1:47</td>
<td>Larry</td>
</tr>
<tr>
<td>2558.mp3</td>
<td>3:06</td>
<td>Jack</td>
</tr>
<tr>
<td>4038.mp3</td>
<td>1:00</td>
<td>Tim</td>
</tr>
<tr>
<td>4040.mp3</td>
<td>1:19</td>
<td>Goodman</td>
</tr>
<tr>
<td>4050.mp3</td>
<td>1:25</td>
<td>Harry</td>
</tr>
<tr>
<td>4174.mp3</td>
<td>2:36</td>
<td>Tommy</td>
</tr>
</tbody>
</table>

Select the music: 4158.mp3
Artist name: Brian
upload the music
# UPLOAD THE MUSIC APPLICATION

**artist Rank verification coin tool**

<table>
<thead>
<tr>
<th>Music</th>
<th>Player</th>
<th>artist name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1450.mp3</td>
<td>2:37</td>
<td>Tim</td>
</tr>
<tr>
<td>2537.mp3</td>
<td>2:43</td>
<td>Tom</td>
</tr>
<tr>
<td>2555.mp3</td>
<td>1:47</td>
<td>Larry</td>
</tr>
<tr>
<td>2558.mp3</td>
<td>3:06</td>
<td>Jack</td>
</tr>
<tr>
<td>4038.mp3</td>
<td>1:00</td>
<td>Tim</td>
</tr>
<tr>
<td>4040.mp3</td>
<td>1:19</td>
<td>Goodman</td>
</tr>
<tr>
<td>4050.mp3</td>
<td>1:25</td>
<td>Harry</td>
</tr>
<tr>
<td>4174.mp3</td>
<td>2:36</td>
<td>Tommy</td>
</tr>
<tr>
<td>4158.mp3</td>
<td>0:48</td>
<td>Brian</td>
</tr>
</tbody>
</table>

Select the music: 

Browse... No file selected.

Artist name

upload the music
GENERATE LISTENING COIN FLOWCHART

1. Pass string
2. Return result
3. Pass transactions
4. Store coin
GENERATE listening COIN APPLICATION

artist Rank verification coin tool

<table>
<thead>
<tr>
<th>Music</th>
<th>Player</th>
<th>artist name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1450.mp3</td>
<td>2:37</td>
<td>Tim</td>
</tr>
<tr>
<td>2537.mp3</td>
<td>2:43</td>
<td>Tom</td>
</tr>
<tr>
<td>2555.mp3</td>
<td>1:47</td>
<td>Larry</td>
</tr>
<tr>
<td>2558.mp3</td>
<td>3:06</td>
<td>Jack</td>
</tr>
<tr>
<td>4038.mp3</td>
<td>1:00</td>
<td>Tim</td>
</tr>
<tr>
<td>4040.mp3</td>
<td>1:19</td>
<td>Goodman</td>
</tr>
<tr>
<td>4050.mp3</td>
<td>1:25</td>
<td>Harry</td>
</tr>
<tr>
<td>4174.mp3</td>
<td>2:36</td>
<td>Tommy</td>
</tr>
<tr>
<td>4158.mp3</td>
<td>0:48</td>
<td>Brian</td>
</tr>
</tbody>
</table>

Select the music: Browse... No file selected.
Artist name
upload the music
TRANSACTION FORMAT

- (artistname)(music_data)(timestamp)(use rIPaddress)(1)(nonce)

- base64_encode($string)

- **Larry**8eac221e13834defb2e14d636e1a2417b30009dee009a6a07dd7b862c1b579b02014-12-0915:59:1512ca17b49af2289436f303e0166030a21e525d266e209267433801a8fd4071a01 n
Web Worker

- New in HTML5

- JavaScript running in the background, without affecting the performance of the page

- Allows the browser to mine for listening coins while listening to music in parallel
WEB WORKER EXAMPLE CODE

```javascript
if(typeof(Worker) !== "undefined") {
    if(typeof(w) === "undefined") {

        w = new Worker("hash_workers2.js");

    }

    w.onmessage = function(event) {
        zeros = event.data;
        var hashString = ""+event.data;
        countNumber = event.data;
    };

    w.postMessage(a);
}
```
WEB WORKER DESIGN

1. `w.postMessage(\text{pass the string})`

2. `w.onmessage(\text{return the result})`
1. Navigate to

2. Get query

3. Pass back data

Web Front End → Rank → Database
# Rank

<table>
<thead>
<tr>
<th>Rank</th>
<th>Artist Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Goodman</td>
</tr>
<tr>
<td>2</td>
<td>Tim</td>
</tr>
<tr>
<td>3</td>
<td>Larry</td>
</tr>
<tr>
<td>4</td>
<td>Tom</td>
</tr>
<tr>
<td>5</td>
<td>Jack</td>
</tr>
</tbody>
</table>

**Artist Name**  
Larry

**Coin**  
Larry8eac221e13834defb2e14d636e1a2417b30009dee009a6a07dd7b862c1b579b02014-12-03 04:00:5712ca17b49af2289436f303e0166030a21e525d266e209267433801a8fd4071a01

**Coin**  
Larry8eac221e13834defb2e14d636e1a2417b30009dee009a6a07dd7b862c1b579b02014-12-03 04:00:5712ca17b49af2289436f303e0166030a21e525d266e209267433801a8fd4071a02

**Coin**  
Larry8eac221e13834defb2e14d636e1a2417b30009dee009a6a07dd7b862c1b579b02014-12-03 04:00:5712ca17b49af2289436f303e0166030a21e525d266e209267433801a8fd4071a03

**Base64**  
TGFycnk4ZWfJMjIxZTEzODM0ZGVmYjJlMTRkN
VERIFICATION TOOL FLOWCHART

1. Navigate to Verification
2. Get query
3. Pass back data
VERIFICATION TOOL APPLICATION

back to music menu

Input Your Hashcoin Value

Text to hash
Calculate

Result:

Input Your Base64 Value

Text to hash
Calculate

Result:

To verify all the coins artist has please enter his name:

Search
VERIFICATION TOOL

Input Your Base64 Value

Text to hash: R29vZG1hbjYzM2RIYTJiNGUwMjBmNDIh
Calculate: MY-SHA256

Result: 000054f5e8501e85a8922c5f6f7802ea7b1084de6b42110e245d1832e53cad39
VERIFICATION TOOL

To verify all the coins artist has please enter his name:

Jack

![Search Button]

<table>
<thead>
<tr>
<th>Artist Name</th>
<th>Coin</th>
<th>verify</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jack</td>
<td>Jack0e2371b4fd4ec412c8688bbf3f40404d4a08686d98458ce032014-12-03 04:36:5812ca17b9af2289436f303e0166030a21e525d266e209267433801a8fd4071a01</td>
<td>000058f01b231b972e338616239e1923b9ba24c126fcb3a3b3f30e86ebf7dcf5</td>
</tr>
<tr>
<td>Jack</td>
<td>Jack0e2371b4fd4ec412c8688bbf3f40404d4a08686d98458ce032014-12-03 13:36:3112ca17b9af2289436f303e0166030a21e525d266e209267433801a8fd4071a01</td>
<td>0000bc142611908988aa6fa6f2eb91b41f66f195bf43bac2c5e46903598124</td>
</tr>
<tr>
<td>Jack</td>
<td>Jack0e2371b4fd4ec412c8688bbf3f40404d4a08686d98458ce032014-12-03 13:36:3112ca17b9af2289436f303e0166030a21e525d266e209267433801a8fd4071a02</td>
<td>0000e440217e6f4869877efb1e1bb2eb0042d69f18b7133f0585f38de8023f4c</td>
</tr>
<tr>
<td>Jack</td>
<td>Jack0e2371b4fd4ec412c8688bbf3f40404d4a08686d98458ce032014-12-03 13:36:3112ca17b9af2289436f303e0166030a21e525d266e209267433801a8fd4071a03</td>
<td>0000236e52d3512d419d007697bf5a3aa7bc6f8ce20111c8038e61e06d6e99ec</td>
</tr>
</tbody>
</table>
ISSUES ENCOUNTERED DURING TESTING

- **WebCL**
  - Interface that allows JavaScript to run code on the GPU
  - Utilize the GPU for processing instead of the CPU
  - Multi-core CPU parallel processing from within a Web browser
  - Out of memory
  - No cross-platform compatibility
WEBCL DESIGN

1. Pass integer

2. Return result

JavaScript

OpenCL
IN OPENCL

- Execute SHA256

- Return result
  - 3 leading zero if less than decimal 1048576
  - 4 leading zero if less than decimal 65536
  - 5 leading zero if less than decimal 4096
  - 6 leading zero if less than decimal 256
ISSUES ENCOUNTERED DURING TESTING

- **Browser**
  - Some ASCII characters not supported
    - Base64 used as solution

Input Your Hashcoin Value

Text to hash: `5d266e209267433801a8fd4071a012284SN`
Calculate: `MY-SHA256`

Result: `4de667d3717f76f61057dc34c3ec545540e1e09917d2b49844aca32458462b37`
CONCLUSION

- Generates listening coins to the artists while one of their songs is being streamed.
- The longer a song is played, the more listening coins are paid out to the artist of the song.
- Allows artists to verify other artists’ listening coins.