BetCoin: Secure Online Sports Betting

ebrad, samcan45, cmauck10, sgsalvas, kalyons

6.857 Spring 2020

1 Introduction

Not only has sports become an integral part of modern entertainment, but wagering on sports has grown immensely in popularity over recent years. More and more people have become interested in betting on all different types of sports. DraftKings is a leading application that has allowed people all over the United States to legally and easily bet on all types of sporting events.

Here, we analyze the security of a popular sports betting website. With thousands of transactions occurring daily between users, their backend, and financial institutions, there are plenty of opportunities for private and sensitive information to be compromised. These companies are required to protect this personal data through their security policy and encryption schemes. Using publicly available knowledge, we will address these policies, establish vulnerabilities, and propose possible solutions.

Some features of sports betting websites include identity and location verification of users. Users also use a credit card, bank account, PayPal, etc. to deposit money into the app. Then users place bets and money transfers occur between the sports betting website and the user. All of these features and transactions must be secure for a sports betting website to operate.

Based on these initial observations regarding a sports betting platform like DraftKings, we would like to provide an evaluation and establish some security procedures. We apply key cryptographic primitives from the course, including digital signatures to validate the identity of users, both public and symmetric key cryptography to validate bets, among others. We will evaluate these formulations against common use cases and see how they stack up to potential attacks by clients. While we do not plan to try to penetrate any existing systems, we will still use public knowledge about DraftKings to evaluate its security procedures and see if we can come up with a similar design.

1.1 Legal Background

In May of 2018, the United States Supreme Court ruled that The Professional and Amateur Sports Protection Act (PASPA) was unconstitutional. PASPA was a federal law that banned sports gambling in the United States outside of Nevada. When the Supreme Court overturned PASPA, it lifted the federal ban on sports gambling. This ruling, however, did not legalize sports gambling, it just left the legality of sports betting up to the individual states rather than the federal government. The ruling now allows each state to make their own laws regarding sports gambling. Since the 2018 ruling, 18 states have legalized some form of sports betting and 5 more states have passed legislation that has not come into effect yet. Each state, however, has different laws about what kind of sports betting is legal and what is not. There are many different laws among the states about where the sports betting can be done and how. All the differences in these laws make it difficult for national firms, such as DraftKings and FanDuel, to follow every law.

Of the 18 states that have legalized sports betting, only 14 have allowed **online** sports betting. In the 14 states that do allow online sports gambling, the laws widely vary. Below we will briefly explain how users can bet online in those states where it is legal.

- Nevada and Illinois require users to register for an online sports betting account in person at a licensed casino. After they sign up for they account, they can then bet through that casino anywhere in the state.
- Iowa is similar to Illinois and Nevada as users have to sign up in person at a casino to be able to bet online. Once users are registered in Iowa, they can place bets on any mobile or online application that is available in the state.
- Mississippi only allows online sports betting while on casino property and the betting must be done through the casino.
- Oregon and Rhode Island's online sports betting is run through the state's lottery. Users must download their state-run app to place bets as it is the only option.
- New Hampshire legalized online sports betting and created a partnership with DraftKings to run the state's online betting. Users in this state can only use DraftKings to place bets.
- Indiana, Pennsylvania, New Jersey, West Virginia and Colorado all have the least strict regulations. These states allow for online sports betting anywhere in the state and users don't have to register with a casino or the state. Users must be 21 years or older and be physically located in the state to place bets in these states. Users must then create an account with one the sites and then they are allowed to bet. These states also allow private firms like DraftKings, FanDuel, and others to enter the market as long as they get licensed by the state and pay taxes.
- Michigan and Tennessee have legalized online sports betting but the law has not taken affect yet. Both are expected to have less strict regulations that allow users to use any site to bet.

One thing we should note is that the **firms** are required to enforce and comply with the laws, not the users. This means the firms, like DraftKings, must verify that a user is 21 or older, so they have to collect sensitive personal information which sometimes includes a person's social security number. Firms also must geo-locate a user to verify a person's location in the state. This means getting a user's specific location on the map to verify their location. Requiring firms to meet all regulations means the firms servers and communication must be secure because they are collecting sensitive personal information as well as tracking users.

1.2 Project Scope and Goals

Our goal is to explore the security policy of common sports betting websites, look for vulnerabilities, and design BetCoin to solve those. We are not redesigning DraftKings or fixing its problems. We are designing our own system and loosely using DraftKings as a model. As far as legal regulations, we are assuming that our site is operating in the five states where any firm can enter the market as long as they meet state regulations. These states are Indiana, Pennsylvania, West Virgina, New Jersey, and Colorado.

2 Definitions

Here, we define several key terms and concepts that are critical to building BetCoin. Our later section with our security policy will define the allowed behaviors for these entities; for now, we just define them to allow for shared understanding.

2.1 Definitions

2.1.1 Users

A user is any minimum access person on the site with an account allowing them to retrieve information on the site as well as undergo basic actions such as place bets and deposit/withdraw funds. A user can have access to the sports betting platform with specific access levels, as defined by the user roles shown below in the security policy.

2.1.2 Event/Contest

An event or contest within the sports betting platform represents a single, abstract notion of a "sporting event." This can range any sport, competition that is played all across the world, no matter how popular or obscure. This definition can also expand to groups of competitions in a tournament, or "pool" style.

An example of an event would be:

Date	Sport	*Team/Player 1	Team/Player 2
9/2/2019	Football	Kansas City Chiefs	San Francisco 49ers

*Note that there are certain events that may not be 1v1 but many different opponents competing against each other. An example would be the entire NCAA March Madness Tournament as an event.

2.1.3 Bet

A bet is a denomination of money placed on a desired outcome of an event. Information involved in a bet can include but is not limited to: date, time, sport, event, bet amount, the spread, odds of various outcomes, desired outcome, and payoff of win. An example of something to bet on can seen below:

Date	Sport	Team/Player 1	Team/Player 2	Odds
9/2/2019	Football	Kansas City (-3.5*)	San Francisco (+3.5*)	(-110**, +110**)

*Here, the (-3.5) and (+3.5) is an example of the "spread" which just means that the amount is either added or subtracted from that teams final score and then compared to the other teams score. For a bet in this case on Kansas City, one would be predicting that Kansas City would beat San Francisco by greater than 3.5 points.

**(-110,+100) represent the odds and therefore the payoffs of a winning bet. A negative value represents how much someone has to bet in order to win $100 and a positive number is how much one wins by betting $100.

There are different types of bets offered with sports betting. The most common are spread, moneyline, total line, and a parlay.

Spread: As shown above, a spread is assigned to each event by BetCoin and the user will bet on the outcome given the spread. In the above example, Kansas City would have to win by greater than 3.5 points to cover the spread and win a bet placed on Kansas City.

Moneyline: A moneyline bet is a bet placed on the outcome of the game regardless of the spread. If a user thought Kansas City would win the game, they can place a moneyline bet on Kansas City and would win a payoff according to the moneyline odds. Often times these odds are different between moneyline and spread bets.

Total Line: A total line bet is placed on the predicted total score of a game. The bet is typically on whether the total score of the game will be “over” or “under” a given amount. Like spread and moneyline, these bets often have their own odds associated with the bet.

Parlay: Finally, a parlay is a grouping of two or more of these types of bets. As bets are aggregated together the odds are adjusted and the payout becomes much larger. However, for a parlay to win, all bets in a parlay must win or the whole parlay is lost and no money is won.

2.2 Transactions

A transaction is a transfer of money between a user and their account. When users want to place bets, they have to deposit money into their BetCoin account. They are also allowed to withdraw money from their account. Users can use bank accounts, credit/debit card cards, PayPal, etc.. to deposit money.

2.2.1 Payment Processor

A payment processor is a financial company that handles transactions and facilitates the secure transfer of money from the customer bank to the merchant bank. Once a payment is verified and authorized, the payment processor sets up the secure link between the two banks to handle the withdrawal of funds from one account and the deposit into the other. Payment processors are used for all merchants to handle credit and debit card payments.

2.2.2 Payment Gateway

A payment gateway facilitates online payments. This is a company that creates a secure connection between the merchant website, where the payment information is entered, and the payment processor that handles the transaction. This secure connection is used to encrypt credit card payment data for every transaction, verifying the authenticity of a transaction and keeping sensitive information secure. There are cases where companies operate as both a payment gateway and payment processor and facilitate both the verification of a transaction as well as the actual transaction itself. This is the case for PayPal, for example.

2.3 Authentication

Each account is associated with a username and password. We also use two-factor authentication to provide an additional layer of security. An even more critical piece of this scheme is the ability to authenticate the true human identity of each user. To do this, we require that each user input their home address, date of birth, legal name, e-mail, phone number, and other identifying information if necessary.

2.3.1 Account Creation

When a user creates an account, the user is required to provide the key identifying information described above. Our system will validate this information to confirm the user’s human identity. Account creation will be discussed more in depth in section 5.1.

2.4 Logging In

Each time a user wants to authenticate into the system, they are required to provide a username and password, as well as have a verifiable, auditable geographic location. This is critical to ensure they are placing bets in a legal location.

3 Vulnerabilities

In a complex system like DraftKings, with millions of transactions occurring, it is essential to establish possible vulnerabilities and attack vectors before they are exploited. Although we are taking a look at DraftKings in particular, these vulnerabilities are designed to be assumptions that could exist on any sports betting platform with elements such as those listed in the definitions section. Without the use of technical penetrative attacks, it is not possible to find actual vulnerabilities. Instead, we will define and discuss some assumptions that can be made given the nature of sports betting.

3.1 Privacy and Property

Within a system like DK, in order for the betting to occur, there exists a voluntary transfer of personal information and property between the provider and the client. It is quite easy to take a brief look and notice that personal details like legal name and bank details, as well as property like money, are necessary for the service to be provided. With sensitive data and property being exchanged, it immediately becomes a vulnerability for adversaries to exploit.

3.1.1 Privacy

As mentioned above, when clients use a platform like DK, personal information must be exchanged for reasons of identification, legality, and payment. With sensitive data being exchanged, this adds necessity for security to ensure that personal data is secured and not accessible to adversaries. In today's day and age, personal data is often targeted by adversaries for monetary gain. If hackers are able to gain private information on users, they are able to sell this private information to buyers across the world. This not only does untold damage to the individuals whose private information is now public, but also deteriorates the public's trust and public image of the service provider. A serious data breach of private information also brings lawsuits which sometimes means significant financial loss to the company. Although DK itself has never been subject to a major data breach, occurrences to other companies have made it very clear that personal data is of utmost importance to protect. All of these factors combined make this aspect of these platforms a serious vulnerability for adversaries to take advantage of if not properly protected.

3.1.2 Property

Just as clients are required to provide sensitive personal details, they are also required to engage in a transfer of property, usually in the form of currency, to and sometimes from the betting platform. In order to place a bet, a bettor must have sufficient funds in their account to cover the desired bet amount. Also, if the bettor wins the bet, they are then entitled to those winnings as they become their property. In any system with money involved, especially on a platform such as sports betting, there exists plentiful opportunities for adversaries to steal, intercept, and commit fraud for personal gain. This is a major vulnerability for this type of platform, as it creates incentive for theft of personal property, which would have detrimental effects on the both the user base as well as the betting platform.

3.2 Technical

In a sports betting setting, adversaries are not just the clients that use the platform. Just as clients can possibly abuse the system, automated attacks and computerized schemes from third party adversaries can be used to compromise the system. We will now discuss some of these attacks that are common for platforms of this nature.

3.2.1 Automated

A site like DK depends on latency to conduct business. When a live event is happening, the "book" (i.e. global bet library) is updating constantly with the newly calculated odds of all events currently happening. When the updates occur, bettors must be updated in a timely manner to ensure fair competition. This is an area where an attack that affects latency and throughput, such as a DDoS attack, could be detrimental to the platform. In this attack, in the worst cases carried out by botnets, sends large amounts of packets to the host, in this case DK, in efforts to block actual client requests from being served. If done properly with enough computational power, it can cause site outages ranging from a few minutes to over an hour. This vulnerability would be detrimental during a live event, as incoming and outgoing requests for bets or betting odds would be blocked by malicious traffic.

As for specifics, DK claims they use SSL to secure all client, server transactions. Although this may mean they actually use TLS which is the newer version, exploits exist for previous versions of SSL such as SSL 3.0. Most commonly, if two parties cannot agree on the newer version, they sometimes revert to SSL 3.0, which is where this attack comes into play. With only roughly 256 queries, an adversary can produce one byte of encrypted plain text. Although DK probably uses the most recent version, this vulnerability could still possibly exist which provides an opportunity for a violation of both privacy and property. When it comes to our BetCoin implementation with HTTPS/SSL, we make sure to use the latest version at all times.

4 BetCoin - Security Policy

Based on the vulnerabilities described above and existing technologies in the field, we propose some new solutions in our system BetCoin. In defining our system, we first outline the security policy here in this section, where we describe the permissible actions and desired behaviors for the various agents involved (i.e. users, servers, networks, potential adversaries, etc.). Then, we provide our technical implementation to several key subsystems in the following section.

Here is the security policy for our system.

4.1 Users and Accounts

4.1.1 Account Owner

An account owner, or "user", must be located in a region where the betting is legal, as determined by government and company regulations. Each account owner is determined by a unique identifier string, which we will refer to as the user ID.

An account owner should be able to view all the available bets in the global bet library, with their associated payouts and odds. But, an account owner should not be able to modify these values at any time. They should have read-only access to the global bet library.

Account owners should have to go through a rigorous verification process to create an account, and are also required to enable two-factor authentication for logging in, which provides an added level of security.

4.1.2 Admin

There are different security levels for different types of admins. Some may only want to view and aggregate data, while others may be able to directly modify it.

For example, an administrator in the system could want to pull a report with aggregate statistics regarding bets placed for a particular event, or in a particular region. The contents of this report must not reveal any individual betting behaviors for users on the platform.

Administrators have the ability to manually set and update the odds and payouts for events. This access must be extremely limited and every update must be logged for auditing purposes.

4.1.3 Developers

Developers have critical access to the code and systems that make the betting infrastructure operate. They should have no access to production databases and servers that contain sensitive user betting data - only administrative users should be able to access these resources.

4.1.4 3rd Party Companies

There are also 3rd party users and companies who have access to view aggregated statistics about account owners and the bets they are placing on the platform. They should only be able to view content approved by administrators, which does not reveal any underlying user betting behavior.

These third parties should not be able to influence any betting results or payouts on the platform, as well as any betting odds associated to particular events. They should have read-only access to aggregate results after events are completed, as to prevent any tampering with bets or automated attacks on the system.

4.2 Placing Bets

As described in the definitions above, placing a bet is the key action associated with our system. Here, we formally define what this action entails and the security notions associated with it.

A user u places a bet b on an event/contest e. There are many different types of bets that b could take on, in a literal sense (i.e. "moneyline", "total line", "spread", etc.) These various specific bet types are defined above. Abstractly, though, let us suppose that b relies on some predicate P on the event e. $p(e)$ is either true or false.

For example, if e is a football game, $P(e)$ could be true if team A beats team B, and false otherwise. Predicates can also become much more complicated and nuanced though, like "team A beats team B by more than n points." But, let us just consider the predicate P as defining the bet.

There is a probability p associated with P being true, and probability $q = 1 - p$ associated with P being false. Therefore, the "odds" of P being true is well-defined as $\frac{p}{1-p} = \frac{p}{q}$. The way in which these odds are manifested in a sports betting platform is a bit more complex, though.

p and q are communicated through the payout options of a bet. Again, there are many nuances here when it comes to betting, but let us start with the most simple bet type, where a user can pay x to the platform, and receive y if and only if P evaluates to true. In this case, $p = \frac{x}{x+y}, q = 1 - p$. x is often set to 100 USD in domestic practice, to keep things standard (i.e. pay $100 to win y).

At any given time t, for any given event e and associated predicate $P(e)$, p should be fixed and immutable. p should only be mutable at a later time $t + \epsilon$ via a secure change process within the system, that is signed, verified and auditable.

4.3 Location Verification

Users should only be able to place bets in legal regions where the government allows such activity to occur. The platform is responsible for maintaining a set of current legal betting locations and enforcing those for users. Users can still log into the platform in any location; the act of placing a bet is what must be geographically verified. The system should also make a best effort to prevent against location fraud and account sharing, where appropriate.

4.4 Financial Transactions

A key component of the sports betting system is the ability to exchange currency to place bets and receive currency as a result of winning bets. This is what makes the system so critical for security - the high volume of monetary value flowing through it at all times.

Before a user can place their first bet, they must successfully link a payment method. A payment method is one of a a debit card, a credit card, or a PayPal account. The system should complete some form of verification to ensure that this a valid payment method.

Users should be able to deposit funds into a balance associated with their unique account, place bets with that balance, and withdraw funds from that balance at any time. The system can also securely provide funds into a user's balance, to encourage more betting. This action, like that of changing a bet, should only be executable via a secure process within the system, that is signed, verified and auditable.

5 Our System Implementation

With a solid understanding of the existing landscape of online sports betting, and a reasonable security policy for a system, we offer our theoretical, cryptographic implementation of BetCoin, a more sercure online sports betting platform. We walk through the technical choices made to design four key subcomponents: account setup and verification, bet encryption, location tracking, and financial transfers. We also provide justification for these choices and how they serve as protection against aforementioned vulnerabilities.

5.1 Account Setup and Verification

To meet state regulations and prevent fraud, we must verify every user's identity that wishes to make an account. In all states, users must be over 21 years old or older to legally place sports bets. Users also cannot be on any federal watch-lists related to fraud or similar crimes. In order to enforce this, we will need strict identity verification, which means asking for sensitive personal information. At a minimum, we will ask for name, birthday, home address, phone number and email address. We will then send this information to a trusted and secure Know Your Customer (KYC) aggregator compliant with U.S. laws to verify a user's identity. If they can't verify a person's identity with that information, we will ask for driver's license number and social security number and retry verifying that person's identity.

Since we are collecting sensitive personal information and sending it to a third party, we must ensure that our system can handle the data properly. First, we will establish a secure connection via HTTPS with the user and our server. This way our server and the user can securely send information to each other. The user will send their account sign up information to our server as a ciphertext via HTTPS. The ciphertext includes the encrypted information and a MAC. We will then take this information and decrypt it. We then establish a secure HTTPS connection with the third-party KYC aggregator and our server. We then repeat the same process to send the user's information to the KYC aggregator. The KYC aggregator then checks the user's identity and sends a response back to our server. (A more detailed description of how the HTTPS connection and encryption would work is shown below in section 5.2).

There are three possible responses we could receive from the identity verification. One response is the aggregator verified the person's identity and they are not eligible for an account, i.e. they are underage or barred from gambling due to other reasons specified by U.S. law and state regulations. The second response could be that the aggregator could not verify the user's identity with the given information. If this happens, we will ask the user for more information and then repeat this process. If the user's identity still cannot verified after five attempts, we will tell the user that we cannot verify their information and cannot create an account for them. The third response is that the user's personal information was verified and we can register their account.

Once we have verified the user's identity and they are eligible for an account, we can officially create their account. To create a user's account, we will create their user ID, which will be linked to their account and their information. We will also need to store their password on our database so the user can log in. We will store the hash of the password and a salt. A salt is random data that is added to the end of the password before hashing it. The actual data we would store would be $h(password|salt)$ where h is a secure cryptographic hash function MD5.

In case we need to contact the user for any reason or need them to reverify their account, we will store the personal information in our database. We will encrypt their name, email, phone number, and home address and store the encrypted data in our database. If we ever needed to contact them, we would decrypt the data and contact the user. To securely reverify user's accounts, we will store the last 4 digits of the user's social security number the same way we store their password. That way we, and a hacker, will not know their actual social security number, just the hash of it. By storing the hash of the last 4 digits of the social security number and a salt, an adversary who gains access to this information would not be able to find out true value of the last 4 of the social security number.

We only encrypt the contact information of the users because we might need to access it to contact users. If we stored it by hashing it, we would not be able to retrieve this information because hash functions are one-way. If an adversary got access to this information, they could possibly decrypt it and obtain the personal information. This information, however, is usually public information and is easier to obtain through a google search rather than hacking a database. Social security numbers are not public information and that is why we store the hash of them. All secret keys for decryption are stored in secure memory in the server hardware.

5.2 Bet Encryption and Security

As described in the security policy above, it is critical that only authorized administrative process can change bet values, and account owning users cannot change bet values once they send a bet with particular probability p of occurring. In this section, we outline the BetCoin scheme for security against both of these vulnerabilities.

First, let us define how our system keeps track of bets. Recall from our definition above that bet b is related to event e via predicate P, with given probability p of occurring, from which the payout ratio $\frac{y}{x}$ can be computed. We use a secure database solution here to manage this data. Each bet b has an associated bet ID, event ID, and predicate ID, as well as p value. There is also a hash h of the current bet value, which is computed as the SHA-512 hash of concatenation of the bet ID, event ID, predicate ID, and p. Each bet entry also has a timestamp at which it is effective. These can all be stored as plaintext in a relational table or other structure. But, write-access to this table must be carefully monitored to prevent unauthorized users or processes from tampering with bet odds and payouts. So, we use a digital signature scheme to secure write access to the `bets` table.

Consider a digital signature scheme $S = (\text{Gen}, \text{Sign}, \text{Ver})$ with security parameter λ. With system initialization, we retrieve $PK, SK \leftarrow \text{Gen}(1^\lambda)$ to get the public and secret key. The secret key is used to sign requests to update a bet entry. Formally, a bet request R includes the bet ID and new probability p' of its occurrence. Then, in order to update a bet probability, the database write request R must be signed via $\sigma = \text{Sign}(SK, R)$. The secret key SK must only be available to secure algorithms running on the system that update event probabilities on some schedule (before or during events, depending on their type). A write request to the database R is only executed if $\text{Ver}(PK, R, \sigma) = 1$. If successful, this writes a new bet entry to the database, rather than clearing old entries. In this way, we store the odds of a bet over time (this will be critical for client validation below). By using this scheme, we ensure that only authorized processes can update event odds, so they could never be tampered with by external parties. We assume that we can use a secure hardware memory solution to prevent leakage of the secret key. We also rotate the public and secret keys on a regular schedule by calling $\text{Gen}(1^\lambda)$.

Now, we must secure the client side of placing bets. Users must only place bets on what is currently a valid set of odds. These bet amounts should be private from other users and those on the network. And, users cannot change the amount placed on a particular bet once it is placed - they can only place a new bet. So, a user u sends a bet b to the server, which contains the following information concatenated: user ID, bet ID, amount placed on the bet, outcome, and timestamp. We also want to add a message authentication code (MAC) to this transfer, to ensure that the value of the bet cannot be tampered with, and came only from u. Let us define the formal procedure:

- The client and server create a secure connection via HTTPS to transfer encrypted data. This provides, abstractly, a shared key k for both the client and server to use for transfering data securely. They must do this symmetric key transfer "twice" to generate k for encryption and k_2 for message authentication. Let us assume that HTTPS provides us with Enc_k and $\text{Dec}_k = \text{Enc}_k^{-1}$, where Enc_k is an ideal block cipher.
- The client sends $c = (\text{Enc}_k(b), \text{MAC}(k_2, \text{Enc}_k(m)))$ across the secure connection.
- The server receives the cipher text and its MAC value, and confirms that the sent MAC value is equal to the MAC of the encrypted bet. If this succeeds, then we know that b is from u and has not been tampered with. Formally, given (c_1, c_2), we confirm that $\text{MAC}(k_2, c_1) = c_2$.

But, we then must confirm that b is actually a valid bet. So, we look up the given bet ID in our database `bets` table and check several properties:

- We find the entry corresponding to this bet at the given timestamp. That is, the first (earliest) entry where the bet ID equals the given bet ID and the effective timestamp is less than or equal to the timestamp provided by the user. Let this found entry be b_d.

- We confirm that the hash of the identified bet b_d from the database equals $h(b)$, where b is the decrypted client bet, and h is a collision-resistant one-way hash function, implemented with SHA-512.
- We confirm that the user has the sufficient funds in their account balance to place b.

If both of these checks succeed, then we can write this entry b to a `placed_bets` table and decrement their account balance by the amount placed on the bet a. The `placed_bets` table should be only modifiable by a secure process (similar to the secure process used to manage the `bets` table above) that can delete stale bets after some time threshold T. Individual bet entries should not be mutable. In this sense, the placed bet table operates as an immutable transaction log of bets over a given recent time window T.

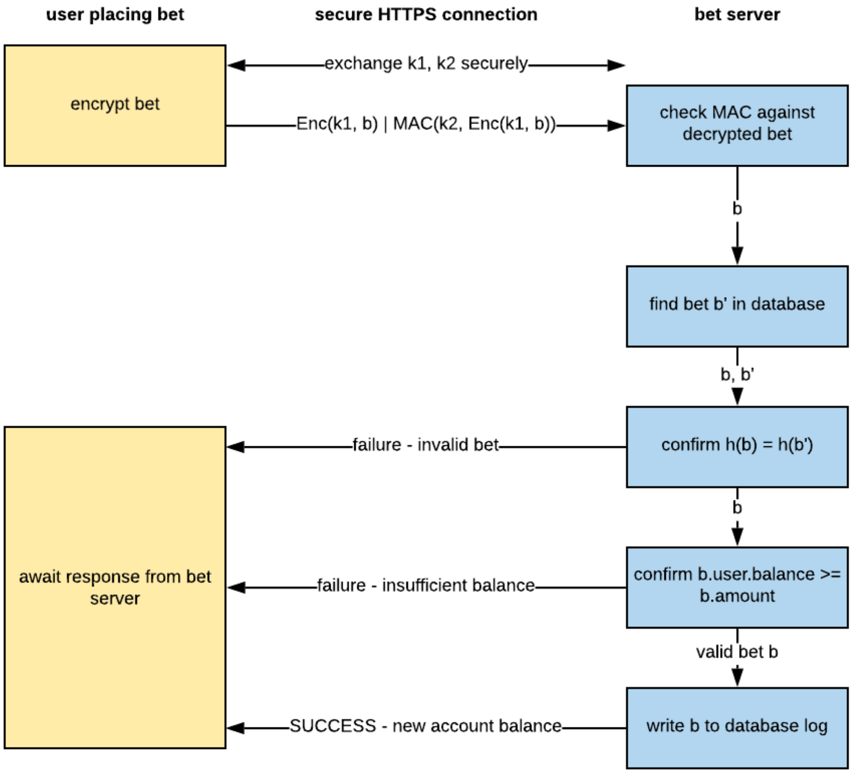

Figure 1. *The high-level design of BetCoin bet encryption verification.*

5.3 Location Tracking

As mentioned previously, sports betting is an opportunity provided to every territory in the United States, but its adoption and platforms are dependent on each territory/state. As a result, for the deployment of BetCoin, we have to ensure that a user ID is accessing our service in a legal and verifiable location.

Upon login and account maintenance (depositing/withdrawing money, updating information, etc.), we follow the protocol detailed in “5.1 Account Setup and Verification” to ensure proper activity. We do not yet have to verify location as the legality issues center around the User ID placing a bet in a legal territory.

Once a user ID wants to place a bet, our service receives an encrypted IP address from the client computer and sends this to GeoComply, which is a third-party service used for verifying locations of computer accesses. There are many third-party services that can be used for this task, but GeoComply seems to be the most popular and reputable given existing commercialized services (they serve nearly 100% of the U.S iGaming market).

GeoComply sends our service a plaintext location from where the user ID is accessing our service. We then check to see if this location resides in a legal area that has the opportunity to use our platform. This is completed using a basic lookup. If this location is not in a legally verified location, we alert the user of this issue and block it from placing any bets. The use of VPNs is a well-known issue for sites attempting to verify locations, so we will also use a third-party service to combat against these VPNs and detect the likelihood of adversaries trying to work their way around the system.

We encrypt this plaintext location concatenated with a time stamp with the hash of the user ID mapped to each of these bet locations. This encryption is accomplished using AES 512 to ensure that we do not store a raw set of sensitive user tracking data. We store the secret key in a secure hardware location and rotate the secret key every day so that it would be nearly impossible for adversaries to crack the code. We will also store the user's most recent bet placement location. This will be used to check if there is any fraud or improper use of trading off the account to another individual. We will have an internal search checking algorithm that calculates the distance between the previous and current zipcodes and evaluates whether the time difference between the two timestamps is reasonable to have traveled that given distance. If our algorithm determines that this distance is infeasible, we mark the action as fraud and send the user a notification indicating this requesting an explanation, otherwise there will be serious consequences for infringing on the legal policy. Each user's previous location will also expire after 24 hours as that information will not be of use to us for detecting fraud going forward. This process can be seen visually below.

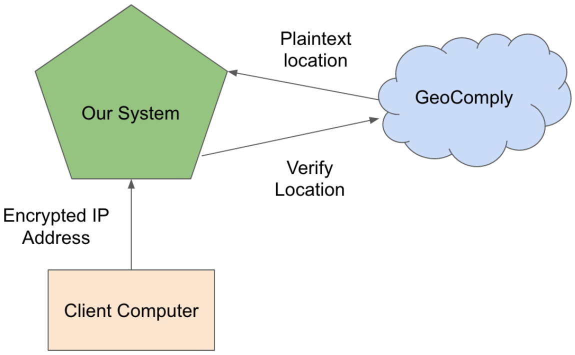

Figure 2. *The high-level design of BetCoin Location Tracking.*

We contemplated storing these locations as a one-way hash, but this process would prove insecure. We decided to not store the fine-grained details of user re-locations within a valid zipcode. We don't need a further level of micromanaging as this would supply more potential issues rather than security (complicate search algorithm for finding fraud). As a result, there is a finite number number of valid locations in the United States that would be valid hashes in which we'd store in our system. This would allow a computationally feasible way for adversaries to figure out the hashing mechanism and unlock the hidden locations.

5.4 Financial Transfers

In order to use the main functions of the site such as placing a bet on an event, a user must first add funds to their account. From here, the account balance must always be managed and verified by the server, and should never be tampered with by anyone except through a secure and verified process.

The first action involved with the use of money on the site is depositing into a user's account. This can be done using a variety of payment methods, including a debit card/credit card, a PayPal account, or a direct Bank Transfer. However, doing this requires a high level of security and identity verification.

When using a debit or credit card, the payment process is straight-forward. A user enters the card information and the amount they want to deposit. The credit card information and payment amount are then processed and authenticated on our own servers, and then encrypted to be sent to payment gateway. The information is encrypted using a Public-Key Encryption Scheme that the payment gateway can then decrypt when received, verify the transaction and send to the payment processor. The payment processor then facilitates the link between the card issuer, the user's bank, and our bank, in order to release the funds to us and therefore charge the user on their statement. Once the payment is received by us from the bank, we move forward with updating the balance of the user. A critical component is once the payment is validated, the payment information is encrypted using AES-512 in CBC mode with a random rotating secret key that is stored separately in a secure hardware location. The encrypted payment information is stored with the hash of the user ID separately on our servers. This way users' payment information is safely secured and private and extremely difficult to crack. It is important that this information not only stays hidden but also difficult to link to a specific person.

The use of PayPal involves the same process for our purposes, however it can be more attractive to users as Paypal acts as its own payment gateway and payment processor that can guarantee secure transactions and verification of the transactions to avoid fraudulent activity.

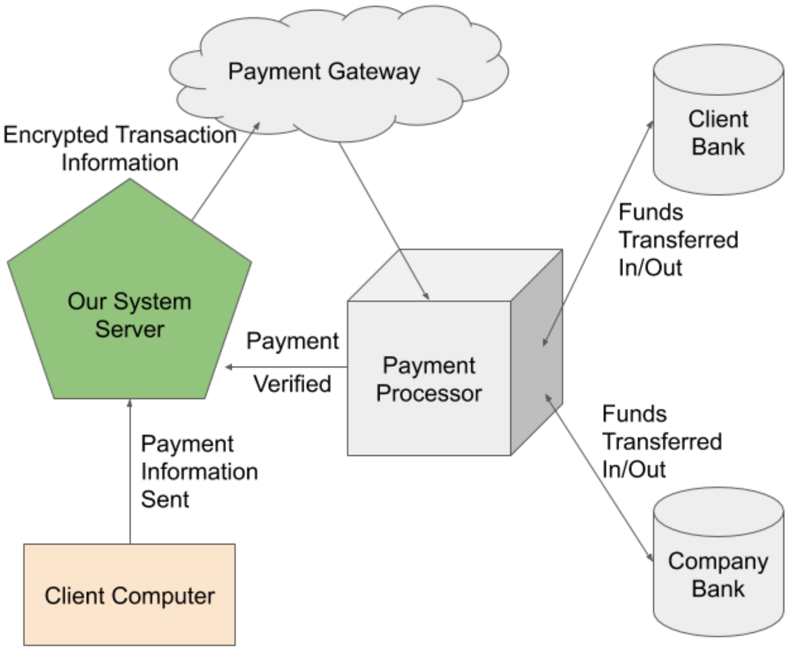

Figure 3. *The high-level design of Money Deposit/Withdrawal Process*

Once the payment is processed and approved, funds can then be released to the user's account. Before this can happen, this information must be encrypted and stored on the central servers. The user ID is first hashed, then concatenated and encrypted with the account balance using a CCA-secure encryption scheme. This encryption is important to make sure that all users' account balance information is privately stored and cannot be changed by any unauthorized users. Once the account balance is stored, it is sent via HTTPS to the user, as described in Section 5.2, and shown to the user. From here, users may place bets in accordance with section 5.2 so long as they have a suitable balance. Changes to a users account balance is never done on the user's end.

When changes to a user's balance need to be made due to the results of placing a bet, the encrypted balance will be pulled and decrypted, the hash of the User ID will be compared to verify the user, and the account balance is then changed to the desired amount and re-encrypted as before to be stored.

Withdrawing funds from your account involves a very similar process to depositing funds. However, first the servers must verify that the amount is compliant with the user's balance. The user establishes a secure connection with the internal servers via HTTPS and sends the amount, a, encrypted as $\mathrm{Enc}_k(a)$ to the server. Once received and then decrypted, the server verifies the amount by locating the user ID hash and decrypting the total balance amount. If the total balance amount is less than or equal to the desired withdrawal amount, a, then the server verifies the transaction and proceeds to the transfer of funds.

Before any payment is processed, the account balance must be updated using the above method. This ensures that while a payment is being processed and money is being transferred, the user cannot place bets with funds he will no longer have. When the transaction is verified by the servers, the account information for the destination of funds is entered by the user and sent encrypted to the payment gateway. The payment gateway then verifies the transaction and securely passes the transaction information to the payment processor. The payment processor sets up the secure link between the user's bank and our bank verifying and facilitating the transfer. The funds are withdrawn from our bank and deposited into the user's account completing the transaction. Once this transaction is complete and validated by our servers, the process is over. If there is an error and the transaction cannot be validated, the user's account is updated to reflect the previous balance, and they are asked to try the transaction again.

6 CIA Security Analysis

Now that we have outlined key subystems in the implementation of BetCoin, we analyze our system's robustness as it relates to the three critical security properties of confidentiality, integrity and availability. Our system design as it stands has prioritized confidentiality and integrity over availability as we have worked more in the realms of protecting user data and avoiding tampering over the network.

6.1 Confidentiality

Confidentiality refers to the protection of secure user data and ensuring that it remains hidden from parties who do not have the correct privileges to view it. Our new BetCoin system works hard to preserve confidentiality in an end-to-end manner. We encrypt all network communications with HTTPS, which uses an underlying symmetric key encryption scheme to protect message contents. We apply this scheme to user IP addresses and locations, financial amounts stored in user account wallets, and bet amounts and odds sent from individual users.

On the server side, we encrypt user locations and all communications with third party services. We use anonymized user ID's when storing bet data so potential adversaries cannot understand who places bets. We encrypt user locations with industry standard encryption algorithms. We protect secret keys with secure memory hardware to ensure that adversaries cannot gain decryption access, and rotate them as needed.

The rationale for placing such a high emphasis on confidentiality is due to the incredible monetary and financial aspect of this service. Users trust BetCoin with an incredible amount of financial value and are attempting to use that value to make more money. This is a very personal action. Also, certain users may want to protect their betting behaviors for other personal or psychological reasons. As a result, confidentiality is a critical component of BetCoin. Our proposed system implementation integrates confidentiality at many levels.

6.2 Integrity

Beyond confidentiality, BetCoin must take integrity into account. Integrity refers to the ability to track and prevent tampering, modification and deletion of data that is managed through the secure system. With the specificity of financial transaction values and multitude of betting options, the ability to tamper with certain values could have dramatic ramifications throughout the system. If a simple bet of $1 is changed to $129 (by switching one bit in an 8-bit integer number), then a user could receive a dramatically different amount of payout based on the outcome of the bet.

When it comes to sending bets, we use a secure MAC scheme to generate message tags that can be validated on the server. This creates a CPA secure mechanism that prevents the ability to forge bets. We also use a collision-resistant ideal hash function on the server side to confirm that a particular bet probability is valid, to prevent odd forging and capturing unwarranted financial gains from the system.

By using trusted and authorized third-party services for location verification and processing payments, we trust the integrity of their subsystems to correctly track users and handle financial transactions. We take a best-effort approach here to maintain integrity and respond to any issues that may arise.

6.3 Availability

This third and final component of the CIA triad, availability, is one that we have prioritized less in our current system implementation. Availability refers to the ability of the system to provide the correct access and resources to users when and where they need them. But, our system does prioritize getting results back to users quickly. For example, by using a third party API for location verification, we can quickly verify whether or not they are valid to login, within an order of seconds. This is critical as users may have an urgency of time when it comes to placing a bet and we want to be available to them as quickly as possible.

Also, our financial subsystem allows for a diverse array of payment options while maintaining security. Users can quickly use whatever method they are most comfortable with and our integrations allow for fast use.

Finally, in terms of placing bets with BetCoin, our system is easily scalable to allow for many requests during popular events (i.e. Super Bowl, World Cup, etc.). We can implement standard techniques for load balancing, database sharding and concurrency to handle peak traffic and maintain availability when users need it most, without compromising security. For the scope of our work here, though, we do not provide an implementation for handling a larger scale attack like DDOS.

7 Future Improvements

As for future improvements, we would like focus on a few things. First, we want to improve handling of volumentric DDoS and botnet attacks that could affect our system. We plan to do this by increasing our server distribution to allow for more computational power and redundancy. Also, but using cloud services instead of in house, some of these attacks would be mitigated. If possible, we would also develop or use a third party service to detect bot-like activity before or soon after it begins in order to mitigate down time as much as possible.

Next, we would like to integrate cryptocurrencies into our platform to provide clients with more flexible payment options. We believe that when implemented properly, cryptocurrencies are a secure and efficient way to exchange monetary value. This introduction would bring its own set of unique challenges, yet we believe the reward is worth it.

Finally, in efforts to keep private information private, we would look into developing a differential privacy system to allow for sharing of user statistics without breaching their privacy. We think that is important to release statistics for both company reputation and user knowledge, but not so much to compromise integrity. With a secure aggregate data algorithm, we know that user data will remain secure while still conveying relevant data through statistics.

8 Conclusion

Overall, BetCoin is a secure and scalable system that could compete with other high-performing services in the online sports betting industry. Confidentiality, Integrity, and Availability are the key principles that guide the design of our system's security and each of its subsystems. Our security policy description primarily focuses on account setup/verification, bet encryption, location tracking, and financial transfers which encompass all of the user's main abilities. We've analyzed the legality, current security policies, and vulnerabilities of existing services to ensure that our platform would provide a safer and better way of betting on sports electronically. The security features of BetCoin allow the service to potentially enter the sports betting market and compete for a share of the multi-billion dollar market.

9 References

Here are some references we have used in compiling information around the current landscape and policies surrounding sports betting.

https://mybettingsites.co.uk/learn/betting-odds-explained/
https://www.legalsportsreport.com/us-betting-sites/
https://www.geocomply.com/
https://www.paypal.com/us/brc/article/how-online-payments-processing-works
https://www.business.com/articles/payment-gateway-vs-payment-processor/
https://www.actionnetwork.com/news/legal-sports-betting-united-states-projections