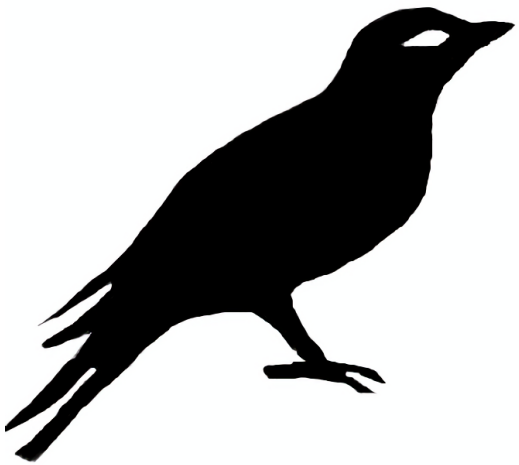
公平なガチャシステム

吉村 優

https://twitter.com/_yyu_
http://qiita.com/yyu
https://github.com/y-yu

July 25, 2016
(Commit ID: b6ff17f)

自己紹介



- 筑波大学 情報科学類卒（学士）

ガチャとは？

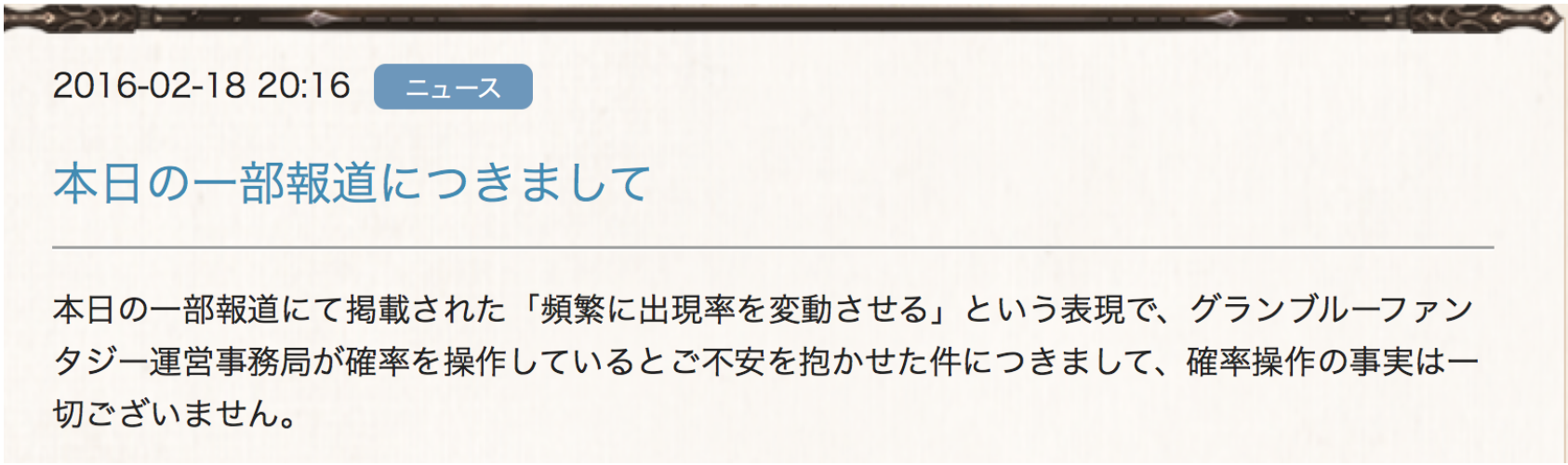
- ソーシャルゲームなどに実装された機能のひとつ
- お金を投入すると**確率**で特定のカードを入手できる

誰が確率を計算するの？

運営のサーバー？

サーバーの実装が正しい確率に従っているのか？

ガチャの確率に対する疑惑



2016-02-18 20:16 ニュース

本日の一部報道につきまして

本日の一部報道にて掲載された「頻繁に出現率を変動させる」という表現で、グランブルーファンタジー運営事務局が確率を操作しているとご不安を抱かせた件につきまして、確率操作の事実は一切ございません。

サーバーで計算される確率に対して疑惑を抱いている人もいる

サーバーで計算される確率を検証するのは無理

クライアントサイドで計算すれば？

リバースエンジニアリングの餌食！

公平なガチャシステム

次のような**公平なガチャシステム**が欲しい

- ユーザーにとっても運営にとってもガチャによるカードの出現確率が実装に基づいて明らかである
- 悪意を持つユーザーや悪意を持つ運営による確率操作ができない

従来の公平なガチャ

- ハッシュ値の衝突に基づくガチャ [1]
- 対称鍵暗号に基づくガチャ [2]

これらを使えばいいのでは？

課題がある！

ハッシュ値の衝突に基づくガチャ [1]

1. サーバーはユーザーにソルト A とデータ C を送信する
2. ユーザーは次を満たすデータ B を探索する*

$$M(K_i) = C \,\&\, \mathrm{Hash}(A \,||\, B)$$

3. ユーザーはデータ B をサーバーへ送信する
4. サーバーはデータを検証し、正しければカード K_i を授与する

課題

- 時間あたりに計算できるハッシュ値の数が、そのままガチャを試行できる回数になる
- 専用ハードウェアを持つ業者など、ハッシュ値の計算能力が高い人が有利である

*& は論理積を意味し、|| は文字列の結合を意味する

対称鍵暗号に基づくガチャ [2]

1. サーバーは値 m を対称鍵で暗号化して、その暗号文 c をユーザーへ送信する
2. ユーザーは任意のカード x を選択しサーバーへ送信する
3. サーバーは c の対称鍵をユーザーへ公開する
4. ユーザーは次のカード k を入手する[†]

$$k := (x + m) \bmod N$$

課題

- サーバーは1 で用いた鍵と、3 で公開する鍵を別にしたとしても、ユーザーはそれを検証できない
- サーバーはカードを操作できるので、サーバーが有利である

[†] N はカードの種類の数である

対称鍵暗号に基づくガチャ [2]

対称鍵暗号では、暗号化に使う鍵と復号に使う鍵が同じと保証できない

やりたいのは情報を与えずに、後からの変更を防ぐこと

コミットメントを使う！

コミットメント [3]

コミットメント

コミット 送信者はコミットしたい情報 b を暗号化して受信者に送信する

公開 送信者は受信者が b を復元できるように付加的な情報を受信者に送信する

- コミットのステップでは、受信者はコミットされた値について何も分からない
- 送信者はコミットのステップ後に、コミットした値を変更することができない

コミットメントを導入したガチャシステム [4]

1. ユーザーは $p = 2q + 1$ となる大きな素数 p, q をランダムに生成して、$\mathbb{Z}_p^*$‡の位数 q の部分群 G から**生成元** $g, v \neq 1$ をランダムに選択して p, q, g, v をサーバーへ送信する
2. サーバーは p, q, g, v を検証し、ランダムに $m \in \{1, \ldots, q-1\}$ を選択し、乱数 $r \in \{1, \ldots, q-1\}$ を用いて $c := g^r v^m \bmod p$ 計算し c をユーザーへ送信する
3. ユーザーはランダムにカードの番号 x を選び、サーバーへ送信する
4. サーバーは r, m を公開する
5. ユーザーは $c \equiv g^r v^m \pmod{p}$ を検証し、次の番号 k に対応するカードを得る

$$k := (m + x) \bmod N$$

---

‡整数 $x = y \bmod p$ かつ $xz \equiv 1 \pmod{p}$ となる y と逆元 z が存在する x の集合である

コミットメントの検証

サーバーが m を後から変更する？

サーバーは m をコミットした後で、$m'(m' \neq m)$ と偽れる

$\Downarrow$ ならば

サーバーは $g^r v^m = g^{r'} v^{m'}$ となる r' を計算できる

$\Downarrow$ ならば

サーバーは g を何乗したら v となるかという**離散対数**が求められる

$$
\begin{aligned}
g^r v^m &= g^{r'} v^{m'} \\
v^{m-m'} &= g^{r'-r} \\
\log_g \left(v^{m-m'}\right) &= r' - r \\
\log_g (v) &= (r' - r)/(m - m')
\end{aligned}
$$

離散対数問題 [5]

定義

$g, p, g^x \bmod p$ が与えられたとき、x を求める問題のことである

g, p が次を満すとき、離散対数問題を現実的な時間内に解くことは困難である

- p は 1024 ビット以上の素数
- $p-1$ の約数の中に、p に近いサイズの素数 q が含まれている
- g が生成元[§]である

[§]生成元は全ての $i = 1, \ldots, q-1$ と $j = 1, \ldots, q-1$ について、$i \neq j$ ならば $g^i \not\equiv g^j \pmod{p}$ となる

コミットメントの検証

サーバーは m をコミットした後で、$m'(m' \neq m)$ と偽れる

$\Downarrow$ ならば

サーバーは g を何乗したら v となるかという**離散対数**が求められる

離散対数問題を解くことは困難であるということに矛盾する

サーバーが m をコミットした後で m' と変更することは困難である

まとめ

- ガチャシステムにおけるサーバーサイドの確率計算に疑念を抱くユーザーがいる
- 従来の公平なガチャには微妙に公平ではない部分がある
- 今回提案したガチャは、コミットメントを用いて完全な公正性を目指した

参考文献

[1] 吉村優.
僕が（ほとんどを）考えた公平なガチャシステム, 2015.

[2] mala（@bulkneets）.
クライアントサイドでガチャ, 2015.

[3] H. デルフス, H. クネーブル.
暗号と確率的アルゴリズム入門 — 数学理論と応用.
シュプリンガーフェアラーク東京, 12 2003.

[4] 吉村優.
コミットメントを用いた公平なガチャシステム, 2016.

[5] 光成滋生.
クラウドを支えるこれからの暗号技術.
秀和システム, 6 2015.

目次

Thank you for listening!
Any question?