Vol.2019-CSEC-87 No.3
2019/12/3

ビットコインにおけるデジタル署名の乱数分析

坂間 潤一郎[1,a)]　金岡 晃[1,b)]

概要：ブロックチェーンは非中央集権性と高い改ざん耐性を提供する技術である。この技術を支えるために暗号技術が盛り込まれており、その 1 つにデジタル署名がある。ブロックチェーンを用いた暗号通貨のビットコインはデジタル署名として ECDSA を採用している。しかし ECDSA は署名時に用いられる乱数に問題がある場合に秘密鍵が他人に知られてしまうという性質がある。ビットコインにおいて秘密鍵の漏洩は資産を他人に盗まれてしまうことと同じである。このような事態を避けるためには正しい手順で乱数生成をすることが必要であるが、実際のトランザクションでは問題のある乱数を使われることがあり、それによる盗難などが既に発生している。本研究はビットコインの署名時に使用された乱数の大規模調査を行い、重複した乱数を使用した署名データを抽出し、その傾向を分析した。同様の調査は他の研究でも行われているが、本研究では既存研究に加え新たな事実を確認することができた。

Random Number Analysis of Digital Signature in Bitcoin.

1. はじめに

ブロックチェーン技術をもとにしているビットコインは、すでに広く利用されており仮想通貨や暗号通貨などの分野を切り拓いてきた。一方で、その応用やトラストの実現などブロックチェーン技術を中心に多くの学術研究がされている。

ビットコインでは高い改ざん耐性の実現にデジタル署名を用いており、署名に用いる秘密鍵は資産の保護に重要な役割を果たす。もし秘密鍵が漏えいすることや計算により導かれてしまう事があれば、資産は第 3 者により自由に移管されてしまうこととなる。時価 580 億円相当の暗号資産 NEM の盗難事件（コインチェック事件）は秘密鍵のずさんな管理により起こされており、秘密鍵の管理は大きな問題となっている。

ビットコインに用いられるデジタル署名 ECDSA では、署名対象のデータに対して秘密鍵で処理を行う際に、乱数が利用される。その乱数の利用が杜撰であれば秘密鍵の計算ができてしまうことが知られている。Courtois らはその攻撃について調査を行い、乱数生成器に問題がある場合の攻撃の実現性を指摘した [2]。その後 Breitner らや Brengel らにより、実際にビットコイントランザクションに含まれる署名の調査が行われ、本来であれば起き得る可能性が非常に低い乱数値の重複が多く見つかっていることが示された [3], [4]。

本研究では、Breitner らや Brengel らと同様に、実際のビットコイントランザクションにおける電子署名データを分析し、その乱数の特徴を調査する。乱数値の重複検知を行うことで、実際のリスクを評価し、これまでの研究との差異を示す。またこれまでの研究では明らかにされていなかった複数の点についても追加調査を行うななど新たな視点での分析結果を提供する。さらに、日々追加されているトランザクションを監視し、過去に発生した乱数値と同じ乱数値の署名を含むトランザクションが生成されたことを検知するシステムを実装し、そのパフォーマンスの評価を行う。

分析調査の結果、既存研究が再現できることが確認でき、また新たに発生した事象を捉えることに成功した。さらに過去に乱数値が重複したトランザクションの送金先アドレスのビットコイン保有を調査し、現在ではそれらトランザクションにリスクがないことを明らかにした。またトランザクション監視システムは十分なパフォーマンスで運用できることを示した。

2. 事前知識

本章では本研究の要となるデジタル署名を用いた秘密鍵

[1] 東邦大学, Toho University
[a)] 651006s@st.toho-u.ac.jp
[b)] akira.kanaoka@is.sci.toho-u.ac.jp

表 1 ECDSA で使用されるパラメータ

値	数値	値の状態
k	乱数	**秘匿**
G	ベースポイント（固定値）	公開
r	$(= kG$ の x 座標 $)$	公開
h	署名するデータのハッシュ値	公開
d	秘密鍵	**秘匿**
p	法とする数（固定値）	公開
s	$(= \frac{h+dr}{k} (\mathrm{mod} p))$	公開

の計算方法及びそれに関連する情報を示す。

2.1 ビットコイントランザクション

トランザクションはビットコインにおいて取引記録の役割を果たす。トランザクションは公開され、誰でも取引内容を閲覧することができる。

トランザクションは様々な情報を保持しているが、ここでは本研究に関係のあるインプットとアウトプットについて解説する。インプットには送り主に関する情報、アウトプットには送り先に関する情報が記載される。トランザクションを生成する送り主はインプット情報として送金金額と資金の保持証明を入力する必要がある。送金金額は自由に記載できてしまうために、その情報の内容を担保するためにデジタル署名が用いられる。

2.2 ECDSA

ECDSA (Elliptic Curve Digital Signature Algorithm) は楕円曲線を使用したデジタル署名アルゴリズムである。ECDSA は標準仕様として SECG (Standards for Efficient Cryptography Group) によって定義されたパラメータが数種類存在するが、ビットコインではその中で secp256k1 が使用される [1]。**表 1** に使用されるパラメータを示す。利用される楕円曲線の位数が 256 ビットの数になっている。また $p = 2^{256} - 2^{32} - 977$ である。

このとき署名値 (r, s) は以下のように求められる。

$$(r, s) = (kG, \frac{h + dr}{k} (\mathrm{mod} p))$$

この署名値はビットコインでは DER エンコードを用いてデータとして表現される。2 つの値はトランザクション内で

<0x30><署名値の長さ><0x02><r の長さ><r の署名値><0x02><s の長さ><s の署名値><0x01>

のように表現される。

2.3 ECDSA の秘密鍵復元

ECDSA はその性質上、同じ秘密鍵、同じ乱数を使用した署名値が 2 つ存在すると秘密鍵が誰にでも計算できてしまう。署名値 1 を $(r, \frac{h_1+dr}{k})$ 署名値 2 を $(r, \frac{h_2+dr}{k})$ とした時、

$$\frac{h_1 - h_2}{\frac{h_1+dr}{k} - \frac{h_2+dr}{k}} = k$$

と計算することで乱数 k が得られる。そして k から秘密鍵 d が得られる。

$$\frac{\frac{h_1+dr}{k} k - h_1}{r} = d$$

よって乱数 k を再利用することや、予測可能な乱数 k を利用することは秘密鍵漏洩の危険性が高いために避けなければならない。

3. 関連研究

本章では本研究の先行研究に当たるものに触れておく。2 章で紹介した場合の他に計算可能な状況についても紹介する。

3.1 Courtois らの研究

Courtois らはビットコインの鍵管理と実際の運用上におけるセキュリティの問題について調査した [2]。そこでは同一の秘密鍵で同一の乱数で複数の署名をしてしまった場合の他に秘密鍵が計算できてしまういくつかの状況が記載されている。

a, b をそれぞれ異なる任意の乱数、k を秘密鍵とする。このとき、以下のケースで秘密鍵が計算可能である。

(1) a が漏洩した場合
(2) a を使い署名した人物が複数いる場合
(3) k で a を使い複数回署名した場合
(4) a, b を使い署名した人物が 2 人いる場合

(2) のケースでは、ある乱数 a を使用して署名した人物が複数いる場合、その人物はもう一方の人物が署名に使った乱数を知っているため秘密鍵を計算することが可能になる。

(4) のケースでは、2 つの乱数 a, b を 2 つの秘密鍵でそれぞれに署名した場合、2 つの乱数 a, b、2 つの秘密鍵を k_1, k_2 とした時に $(a, k_1), (a, k_2), (b, k_1), (b, k_2)$ の組み合わせができる。

この時、

$$s_1 = \frac{H(m_1) + r_a * k_1}{a}$$
$$s_2 = \frac{H(m_2) + r_a * k_2}{a}$$
$$s_3 = \frac{H(m_3) + r_b * k_1}{a}$$
$$s_4 = \frac{H(m_4) + r_b * k_2}{a}$$

と置き、

$$s_2 * k_1 - s_1 * k_2 = \frac{H(m2) * s_1 - H(m_1) * s_2}{r_a}$$
$$s_4 * k_1 - s_3 * k_2 = \frac{H(m4) * s_3 - H(m_3) * s_4}{r_b}$$

これらの式を解くことで k_1, k_2 が得られる。

3.2 Breitner らの研究

Breitner らは格子理論に基づく Hidden Number Problem を ECDSA に適用した手法を提案した。署名時の乱数に偏りがある場合その署名の秘密鍵を計算するためにこの手法を使用することができる。Breitner らは様々な暗号通貨から集めた ECDSA 署名に対してこの手法を用い秘密鍵の計算を行った。それらの結果を暗号通貨ごとにまとめ、さらなる分析を行った。

ビットコインにおいては、2018 年 9 月 13 日（block height 541244）までのトランザクションデータを調査対象とし、そこから 446,605,479 個の異なる鍵と 975,560,082 個の署名を抽出した。そしべそれらのうちうち 40,497,752 個の鍵が複数回署名を生成していたことを明らかにした。

3.3 Brengel らの研究

Brengel らはビットコインの秘密鍵流出について、OSINT に公開されてしまっている情報を元にした明示的流出と署名の不備による暗黙的流出の調査を行った。OSINT で発生する明示的な鍵漏洩については Pastebin を利用した。2017/09 から 2018/03 までの Pastebin フィードの監視により公開されたビットコインの秘密鍵を見つけ、攻撃者が 22.40BTC を盗むことができたこと明らかにした。暗黙的流出については Breitner らの手法と同様に手順に不備のある署名を対象に調査を行った。Brengel らは論文内でこの手法を悪用した攻撃はすでに起きており、412.80BTC を盗むことが可能であると明言した。

4. ビットコインの署名に利用されている乱数の調査

本研究では、他の先行研究と同様に署名値に含まれる r の値の重複を検出することにより、署名時に同じ乱数が使用されているかの調査を行った。調査実施にあたりトランザクションの収集を行い、収集したトランザクションから署名値を抜きだし、トランザクションに含まれる署名値の r の値の重複抽出を行った。ここではその詳細を解説する。

4.1 トランザクションの収集

ビットコインではブロックチェーン検索が可能なエクスプローラーが公開されている。本研究ではビットコインのエクスプローラーを提供している「BLOCKCHAIN [*1]」の WebAPI を利用し、ビットコインのトランザクションのクローリングを行った。これにより JSON 形式のトランザクションのデータを取得し、これらをテキストファイルに保存した。

[*1] https://blockchain.info

4.2 署名値 r の抜き出し

DER エンコードに従いトランザクション内に記載されている署名値の該当部分から r の値を抜き出した。これをテキストファイルに保存し r の値のリストの作成を行った。

4.3 署名値の重複の調査

作成した r の値のリストをソートし、重複を検出した。同時に重複の回数も調査した。

4.4 出現頻度の高い署名値の Web 検索

実際に重複のあった r の値は、なにか特別な理由で利用されている可能性もある。また secp256k1 パラメータ上の特徴のある値となっている可能性もある。これらを踏まえ、重複した r 値自身を検索エンジンを用いて Web 検索をした。先行研究結果の成果として著者らが Web で重複乱数値を公開している Web サイトがあるため、それらを除外し、それ以外に検索結果として意味のあるページが出てきた場合、さらなる調査を行った。

5. 乱数利用の調査結果

本調査では、総数 17,235,188 のトランザクションに対して署名値の抽出を行い、その結果 705,826,061 の r の値を抽出した。1 つのトランザクションに複数の署名を持つ場合があるため、トランザクションの総数と異なる。重複した r の値とその件数の結果を表 2 に示す。分析結果は先行研究と同様の結果が得られた再現の部分と本研究で新たに発見できた部分に分けて記載する。

5.1 既存研究による調査の再現

5.1.1 Breitner らの研究による調査結果

重複した r の値の Web 検索によって、一部の r の値は情報が取得できた。

重複回数が最も多かった r の値は Breitner のの乱数は $k_1 = \frac{(n-1)}{2}$ であることがわかった。ビットコインの ECDSA で使用されるパラメータは secp256k1 によって定義されており、ここでの位数 n は

0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEBAAEDCE6AF48A03BBFD25E8CD0364141

である。

この位数の場合、k_1 による演算の結果 k_1G の x 座標の値が小さい値になることが知られている。k_1 は 166 ビットで $2k_1G = G$ となる。

同様に Web 検索によって 4 番目に多く重複が見られた r の値の乱数も判明した。この値の乱数 k は 1 であることが明らかになった。

5.1.2 Brengel らの研究と同様の結果の部分

本研究で得られた重複している r の値の順位と Brengel

らの論文に記載されていた多く重複が見られた r の値のトップ 10 の順位は同じであることが確認できた。

5.2 本研究での新たな結果

5.2.1 それぞれの r の値の重複度

重複している r の種類は 5,802 件、重複している r の総数は 2,582,952 であることが確認できた。これらの数は既存研究と異なり、正しく検証できていると仮定するとこれまでの研究以降にも重複した乱数が引き続いて発生していることを示している。

重複した乱数が出現するのは全署名値の 0.365%にあたる。また重複が発生した乱数において、重複件数の平均値を求めたところ、445,183 となった。一部の値が非常に大きいため偏りが発生していることがわかる。

5.2.2 出現頻度の高い値の特徴

- 1 位この r 値については先行研究で見つかり調査がなされている。
- 2 位この r 値については 101 種類のアドレスより、5 回の時刻にわけて同時にそれぞれが 76 回出現していることが確認できた。
- 3 位この値は同一アドレス 1 種類のみと確認できた。時間はまばらである。
- 4 位この値についても先行研究で調査されている。

5.2.3 調査対象期間で乱数が十分に均一だった場合の重複発生数の近似値の計算

P256 の位数 $n =$
0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEBAAEDCE6AF48A03BBFD25E8CD0364141

を元にバースデイパラドクスによる重複発生の確率を試算した。その結果は、2.151228252993635217271667505023220073867 6 $\times 10^{-60}$ となった。この値を超える数発生しているものは、乱数に問題があると考えることができる。

5.2.4 r の値の重複があったアドレスの残高

r の値に重複が発生した場合、その署名に利用されたアドレスに保持されている資産は第 3 者に不正に引き出される可能性が強い。そこで、r の重複が発生したトランザクションにおける現時点でのアドレス残高を調査した。

調査の結果、関連するトランザクションはすべて送金済みであり 0BTC であることが確認できた。

5.2.5 既存研究と比較して一部重複署名の回数増加

Brengel らの研究によって調査された r の値の重複が多いものトップ 10 の表と本研究の結果を比較すると 4 番目に多い値の回数が 1 回増えていることが確認できた。また、他の値についても回数が増えているものが散見された。

6. 考察

本研究における結果を踏まえて考察を行う。

6.1 署名値の重複について

結果のバースデイパラドクスによる計算から見て取れるように実際の乱数の重複回数は多いことが確認できる。使用される乱数が十分にランダムである場合、乱数が重複する可能性は極めて小さい。

6.2 署名値の重複が起こる原因について

これらの重複の原因としてまず考えられるのはビットコインの取引を行うソフトウェアの実装の不備である。またこれらのソフトウェアが実装に使用しているライブラリに不備があることも考えられる。いずれにしても署名時の乱数の生成に何か問題があることは言えるだろう。

7. 対策の提案

Breitner らの研究によるとビットコインのデフォルトのライブラリに実装されている deterministic ECDSA を使用することにより署名の不備による秘密鍵の漏洩は防ぐことができると述べられている。しかし本研究の結果から過去に重複が起こった署名として公開されているにも関わらず回数が増えていることが確認できる。一般のユーザーは自分が使っているソフトウェアのライブラリが乱数生成に不備がないか確認することはおそらく無いだろう。そこで我々はこの問題の対策として、送信したトランザクションの署名が過去に使われた乱数を使用していないか確認し通知を行うプログラムを作成した。このプログラムによりビットコインを取り扱う事業者はユーザーの署名の不備を検出した後早急にユーザーに通知を行うことができる。実際に作成したプログラムは過去に重複したことがある署名を検出するプログラムで稼働させ実用に足るものか検証した。20 回の計測したところで新たに送信されたトランザクションの署名の確認を行うのに要した時間は平均 10.755 秒であった。この検証は OS は ubuntu 18.04、CPU は i7-9700K 3.60GHz、RAM は 16GB のマシンで行った。

8. まとめ

分析調査の結果、既存研究が再現できることが確認でき、重複した乱数が引き続いて発生していること、出現頻度の高い値の特徴といった新たに発生した事象を捉えることことができた。また過去に乱数値が重複したトランザクションの送金先アドレスのビットコイン保有を調査しすることにより、それらの残高は 0BTC であり現在はそれらトランザクションには資金流出のリスクがないことを明らかにした。またトランザクション監視システムは新たなトランザ

表 2 それぞれの r の値の重複度

	件数	値 (hex)
1	2,552,152	3b78ce563f89a0ed9414f5aa28ad0d96d6795f9c63
2	7,900	6fcf15e8d272d1a995af6fcc9d6c0c2f4c0b6b0525142e8af866dd8dad4b
3	265	1206589b08a84cb090431daa4f8d18934a20c8fa52ad534c5ba0abb3232be1d9
4	252	79be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798
5	91	2ef0d2ae4c49c37703ba16a3126e27763e124ff3338fb93577ed7bd79ed0d19e
6	83	06cce13d7911baa7856dec8c6358aaa1fb119b5a77d0e4d75d5a61acae05fcfb
7	76	d47ce4c025c35ec440bc81d99834a624875161a26bf56ef7fdc0f5d52f843ad1
8	68	281d3da7518241cd8ee30cd57ae3173a1bd9ee5e3b02a46ba30f25cd5b4c6aa8
9	64	008216f63d28f4dc0b6909a330d2af09b93df9dd3b853958c4d203d530328d8ed1
10	52	5d4eb477760cf19ff00fcb4bab0856de9e1ce7764d829a71d379367684712be4

クションが送信されるとすぐにそれを検知し、次の新たなトランザクションが出現する前に検知したトランザクションの検証を終えることができた。これは運用できる十分なパフォーマンスがあると言える。

参考文献

[1] BROWN, Daniel RL. Sec 2: Recommended elliptic curve domain parameters. Standars for Efficient Cryptography, 2010. https://www.secg.org/sec2-v2.pdf

[2] Courtois, Nicolas T., Filippo Valsorda, and Pinar Emirdag. "Private Key Recovery Combination Attacks: On Extreme Fragility of Popular Bitcoin Key Management, Wallet and Cold Storage Solutions in Presence of Poor RNG Events." (2014).

[3] Breitner, Joachim, and Nadia Heninger. "Biased Nonce Sense: Lattice Attacks against Weak ECDSA Signatures in Cryptocurrencies." IACR Cryptology ePrint Archive 2019 (2019): 23.

[4] Brengel, Michael, and Christian Rossow. "Identifying key leakage of bitcoin users." International Symposium on Research in Attacks, Intrusions, and Defenses. Springer, Cham, 2018.