Non canonical scalars, malleability and wrong signatures stored in the blockchain

dangerousfreedom

September 2, 2022

1 Introduction

This a small report trying to document the "malleable" transactions stored in the blockchain due to a wrong interpretation of non-reduced scalars.

Basically, we have a function called addKeys2 that does $R = aG + bP$ where a, b are scalars, G is the base point and P is a point. The result R of this equation is wrong for certain conditions of the scalar a (being non-reduced with a[31] > 127), which can be misinterpreted as another value a' leading to a malleable transaction as the same scalar could generate two different signatures due to this imprecision in the code.

2 Transactions with wrong signatures

This is (hopefully) the full list of transactions presenting this behaviour.

#	Tx hash	Date
1	0647d386365f6bfd312b0fbe966f5c85f19159ccf9003af8387f332451e6c94c	2017-07-17
2	d5d725e7a76dab7e2ca97d941403936a0fbf5e8874e9ef3becd973e4598a8cb1	2017-07-17
3	991712d33cdff510254ffacfb9e04f9cd54a4f08bf53388c107f72383f03c462	2017-07-18
4	475c008c48520949843a00701fdd15b6db76bdf24b3f635bdaeb9f4a684700f8	2017-07-19
5	2fecc6f9269264fb0f036de586451f4ae0d0c7cec8ace89d93135b90bbdc6263	2017-07-19
6	0aba657035c498037e00d8597eee74b06f6d89c900a74af21fc717bb86002c8d	2017-10-02
7	a6dc6d18d493e9f2749e5d0991b7703b26779d25b286de55b99a72d927d102bb	2017-10-03
8	93c42d7938a6534816e8583e6d4accac1cff200fa4d2b51daddab91456e7f9eb	2017-10-03
9	8171e8ac8154a20d1d345ef5b872ab2b019096efde220c67cce893b1ea25cc76	2017-10-03
10	3f7beae73b73e3b0864a16f9950f5bb80ab7cc74498fc50e747b301b2ecb77a7	2017-10-04
11	d21685edb1b6911b0bbe995749608b84aebb42cd99cb1487dfef4f9a8253eebb	2017-10-05
12	37ef3484d72ac43e14d49cf5d8058607553c2018fc795c1e8dcb39b9d033a1e6	2017-10-05
13	d1ea609d17125f407fa8b80d5fc74add85f12efa242d16509bc9d18a292f641f	2017-10-05
14	3d407ce4a7b7f22f923987f501a7d07dc9fa9bb9568c3c8e1a2c891501597b60	2017-10-05
15	195a9d291477b6d0c95a9c006c42a324a89dc4ffae529bc894b94284ab248359	2017-10-05
16	e9f0234705be100032dcd1e9c631822f00b0a3d7e5748b6e6533e8eec47f2644	2017-10-05
17	cc3ebd46fa3f85d6a2aa444d82afb5331d9bd8d6d53fcea5e6920c448c57d72a	2017-10-05
18	d6187b450e32eb1392b0b5b211cbcb8d66aecaf90705bd1235f1a99de82577dc	2017-10-05
19	7a47b4e8261679fc74a419bf23823b8f25cc74a67509836760e8cb0bc463438b	2017-10-06
20	3037b7c0bdb7019ab45d726c4f0354bdd0b1a025ac88b08752af1fd9b4ca7f81	2017-12-14
21	f5b60a9e22177f6ca68d413ceb39db40d4d43aacddf70191c0dbc3c36be7c588	2017-12-14
22	e4b7982b081a17892525f1b1d3011ec06a0820cbf451d3a64f8ea998104a753c	2018-02-04

Table 1: List of transactions with wrong signatures.

3 Correct signatures

Here is presented the stored value in the blockchain (red) in comparison to the correct value (green) that should be stored if the interpretation of these scalars were correct.

#	Value
1	f64806929fa873d76a10a5471fcffafc45d8ec932456af5dc4b1c7bd9da3bc0d
1	6c5a5479345ea11e8d5f322643b0a7d2cd4ba9372ed89db8784b7611d0e05d09
2	75363d75a40eb4393cf75a8cdc112aeef69a6be8124ea62846c1dff822816d00
2	f6112dd6711ff2c6edb09dc296982b6b7dba0a233f059cf8cee09769c74b3b0c
2	a1dccdd780af65f7f448c43f58cab1bff41c0bdc7697a417ff6c042ef858bc0b
2	6fdf4c48f2ad9a1ff87d6f4c0058aab67275edc70d3b5ce9510b78c595b8950f
3	30503ff05c63159d70290087d08759cebc00d4c9e0a03d282b38d863aa48eb04
3	e6a892d20b30e269cc6c0f38c2c926b24a77e5b08c0047e8a6d4c18f26d76c0b
4	50563a677f9dbdfab3e634b08c5b2a7ba68ac5dcb3745ee4e79986fcbbe65b0f
4	257d854804b489a94cc205197664583eb419105e6e0281c9934a7093b30a8d06
5	3d2694db80cb64365ee1329fc6fd15a0da3e390d38aed2698917116047a44503
5	81eceee886b76fadbee762143a25cb87507d3cbb9135346899dee7ceac1ca80e
6	fafee92bf2e9d2a326c32def30a96cc94d91f06725e45c02b56d36c2ecdff700
6	6c6b4325c99bd170e2f09c5566e1f3b0ca952c55e1d4a566f0ca672bbfd50a06
7	079568a79995ec887f10e42aa50dab5d90a84614129e2632ef00400f4dd3580b
7	ac121f1260b927d23eff2a1069492a06b77022839efa286eb96f1574e90a9107
8	21c036959ec10d60b4ec671e5b880074d173a4716871eb2683a1e1b9448acb0b
8	a22432ba379622a552c05c7b931a19389284d44fbd53302e24a4f3b82f7d1f07
8	ba05389ea6b239c9ab6afd9653bcf23738ea48780b71f0f8d5d06ac55be92708
8	61a741035dc13d42f8d07f3948bdf4645c0541eceae9a5904a5c8d49a914df0f
9	82b9145bfcba88c7de483b93987fef4004d38cca4d23e88bdaac0d6ec682d10b
9	c367f6c98c7f0bc76c30b68c7fe5395e656d2401d873527cb7883d8b4a344008
10	3106ff969bfe4cca8ae078a31ba632c9a833601b25ff0b35b219c7b0e12f6002
10	4cc30ce354700a8cf610ea7f6e724d43bcefae2297cc16da3581f9a02e1ef008
10	b24001cbcf54a1b0cd3d9c08c295a3d29150c3d24efe659d7a44e14bcdf8af09
10	56c94bcd778b9daa674ab1aace5ff8bbee7b7b0e9f2d4e28585d2449d67b3700
11	cc76a20d56e3d9e40b8520dab145d8a7866a4e9099d0398147cb4aff8fbbac0c
11	f99ef6d2430c91b368546e3127324f9862811a949a44b3a2ef54443ec8a5fe0d
11	b77006517fd1d4645f17331132461d2c86eff23d47740691745d15742a64f60d
11	300158c07c5656403ae8a3ce557a489f565321fc49ba2e77e56bed62bfd2c408
12	00dc8e630ed34007acc17f5281ddc56bfe0556455cea12d0172cea343d22bd06
12	43a5d5d80fef3f6750e2e8263074adc587c59ad608c07eac8549e2114260b507
13	6870266cc969fe6be28bf4cb2bcae38eadc51d6bdfb98e3a38463cc9a598c109
13	02b8daafdd4f1b37c7a81c36f210bbbfb7fdbdeec86d8bc9d2d86476415e7c02
13	059f648616691e1cd90e954570bf764fcafafbcc9f1187e9549e5e1dade37907
13	64ac71688ae75c05a5ad7621eaafc8f851ba3cef396433b5142ed938fa49210a
14	98c94e5619539573276c56dcc60af818555793ee0f9408f2b1dd688755747805

14	5a9714b561b7103394116ba90d8fe4f5b754e593eb0e9eb4b3d5897b93c8f401
14	f44121ca3512c50320d6b60bd5bd1e715412ba2595b87cfd6d25d786c619be06
14	8d71204547b95544b89645fdfe12c5828d02ae774cb83b8ae133c844e97ab607
15	396a5230d1ed23719372b634ed288512318f4e5ab7d6e19378ff23869cef7302
15	f66e2d310391b9f886d919237002462c468d48924d6a114cc609410dd1f08d01
16	9fe52679338990a1e4406ddf47166695f5c92b3dc041a252f057386cb6e8df03
16	3353f7ca5591651b7db965e77e2896819653044921e1c35dfb3caacc184cb002
16	782fdb1e79cf4cef8cece25aecfb0b25af0d8ef3ed13f0c0025710d59c346806
16	1b87a795dd6fa66534a3bd581790a5e55c3be43de71c15691b78423f9e8e420e
17	e4e513830681ea66e3b7bab0a712977788025af18d0b7d6b91aff50ce834c102
17	c821ad3a21566e84a1fb114b4659df7b5579fff54fdbf77df4ef64b871c9cd0b
18	0137ed34592743da38e6a20455158f6b5e608301fbf9d1e92af90044650fe20e
18	4a1f3668821d336da20384b286a22fcb9a236631a908d2ebf45d2242a6a7fe07
19	8e3b60c06f58e3e90ae93d27c76561b57393baa1ce486a6ad4854b125d059004
19	44c032e8a1f24fd12f7753930c950812ed6e6f4b02c7687fe55af82a65a3ed03
20	fbe9baa6235c20ccec603caa96486aed4d04049d2e6279f824fc64badc461305
20	5745a83cc93803f2617dd6bef6e36825ef8816a1504e31982bbbd06a1131100b
21	dba5416c5e4600a444b95419a1512d96e48499e3d15ae799507020756c193802
21	65b95744c9193581693023dbde78156a3c00a70ee85717ade0a231c8dac0f90d
22	a29635109c80c54623a8a63e412daec9bc4646fdad9ac5db006d7c541b645302
22	9aff4d99740c18dc2c6d1258ca585933a18d9838ebab98d8fcd51f867a0b990b
22	d255061e3e2ab8b3c7c0b5b730211593abcbaf0ae9438f094b61562577acec0a
22	dcbd69e5ae0cea81a60ed348e79599da68f96b89fd419c36166d50de47f65607

Table 2: List of the wrong signatures stored in the blockchain (red) and the correct ones (green) retrieved from using the proper implementation of the math equations.

On table 3 we find the quantity of wrong scalars for each transaction according to the index from table 1. Totaling 72 scalars.

#	Quantity of wrong scalars	In/Out
1	5	2/2
2	9	2/2
3	4	2/2
4	3	2/2
5	3	2/2
6	3	3/2
7	2	2/2
8	3	2/2
9	2	1/2
10	4	4/2
11	4	2/2

12	2	2/2
13	4	1/2
14	4	2/2
15	2	1/2
16	4	1/2
17	2	3/2
18	2	1/2
19	2	2/2
20	2	2/2
21	2	2/2
22	4	2/2

Table 3: Quantity of wrong scalars at each transaction and the number of inputs and outputs.

4 Wrong scalars

On table 4, some examples are shown from the 72 wrong scalars.

Description	Value
Stored	cb2be144948166d0a9edb831ea586da0c376efa217871505ad77f6ff80f203f8
Reduced	e8c079d208b352a71abd36a5deb45c67c276efa217871505ad77f6ff80f20308
Interpreted	b8ffd6a1aee47828808ab0d4c8524cb5c376efa217871505ad77f6ff80f20308
Stored	343d3df8a1051c15a400649c423dc4ed58bef49c50caef6ca4a618b80dee22f4
Reduced	51d2d585163708ec14d0e10f3799b3b457bef49c50caef6ca4a618b80dee2204
Interpreted	21113355bc682e6d7a9d5b3f2137a30259bef49c50caef6ca4a618b80dee2204
Stored	c14f75d612800ca2c1dcfa387a42c9cc086c005bc94b18d204dd61342418eba7
Reduced	7f08db340ba1543162bc4edbc77f13fc076c005bc94b18d204dd61342418eb07
Interpreted	4f473804b1d27ab2c789c80ab21d034a096c005bc94b18d204dd61342418eb07
Stored	000102030405060708090a0b0c0d0e0f826c4f6e2329a31bc5bc320af0b2bcbb
Reduced	d1e57104e2c23b3ed24b660a7b507929816c4f6e2329a31bc5bc320af0b2bc0b
Interpreted	a124cfd387f461bf3719e03965ee6877826c4f6e2329a31bc5bc320af0b2bc0b

Table 4: Quantity of wrong scalars at each transaction and the number of inputs and outputs.

5 Implications

Although someone knew about this scalar misinterpretation (for example the scalar 0001020 30405060708090a0b0c0d0e0f826c4f6e2329a31bc5bc320af0b2bcbb really looks like human generated) it is not an exploitable issue as briefly demonstrated below. Suppose that we have the following:

- Input commitment: $C_i = xG + aH$
- Output commitment: $C_o = yG + bH$
- Fee Commitment: $C_f = fH$

In order to balance the amounts, we need to have $C_i = C_o + C_f$. Therefore:

$$x = y$$
$$a = b + f$$

as we assume that nobody knows a relation between G and H. Notice that the equation $a = b + f$ is modulo p, where $p = 2^{252} + 27742317777372353535851937790883648493$ is a number of the order 2^{253} and the rangeproofs amounts are of the order $m = 2^{64}$.

Let's consider the following scenarios as a thought experiment:

a. Inflation was generated commiting to a value $v > m$:

   Let's assume that someone could cheat and commit to a value $v = a - f + p$. This is not possible for two reasons: 1) The values are modulos p and $b = a - f = a - f + p = v$. 2) the rangeproofs accepts only 64 bits and it is impossible to write any number $v > 2^{64}$ as a number with 64 bits.

b. Inflation was generated commiting to a value $v < m$ and $v \neq b$

   Let's assume that someone could cheat and commit to a value $v = a - f$ and $v \neq b$. How could that be done?

   The only way I see that to happen is by making the software interpret b as v somehow, which is not the case so far.

c. Someone found a relation between G and H.

   In this case, the blinding factors are no longer separated from the amounts and the equations could cross-talk.

   There is no reason to believe that it is possible, therefore a misinterpretation on the scalars of the blinding factors do not impact the amounts.

6 Conclusion so far

Due to the use NUMS (Nothing-Up-My-Sleeve) points (G and H), we don't know any relation like $G = \alpha H$ to exploit this misinterpretation of the scalars.