THE BLOCKCARD PROTOCOL: IT'S THE PROOF-OF-THOUGHT THAT COUNTS

Robert S. Adlemir
Tree University
Department of Computer Science
rsa@cs.tree.edu

Chris K. Wong
UC Berkeley
Department of Computer Arts
chriskw@berkeley.edu

June 28, 2019

ABSTRACT

We identify a major security flaw in modern gift transaction protocol that allows for malicious entities to send questionable metadata to insecure[1] recipients. To address these weaknesses we introduce the Blockcard protocol, a novel variant of Blockchain technology that uses an asymmetric proof-of-work CPU cost function over payload metadata to provide a cryptographically secure and efficient method of verifying that gift-givers thought enough about the recipients payload or lack thereof for it to count. This has the advantage of making it computationally infeasible and socially awkward for adversarial gift-givers to double-spend, spoof, or precompute their celebratory thoughts.

1 Introduction

In the most recent set of standards for gift transaction protocol it is not uncommon for a gift-giver to send metadata with an empty or completely randomized payload. This functions as an acknowledgement of the recipient's special occasion and is accepted under the current standard outlined in RFC6592[2] [1]. For example, the metadata in Figure 1 can pass through the system without an associated payload.

Figure 1: Gift metadata for empty payload[3]

```
1 Dear Bob,
2
3 Happy belated birthday! I didn't have the chance to get you
4 a gift this year, but hey, it's the thought that counts!
5 I know you've had some challenges to overcome recently, but
6 if anyone can get through it its you!
7
8 Best regards,
9 Alice
```

In these cases, the sender encapsulates the metadata, usually in a nice envelope, and relays it to the recipient without a payload. Which is totally fine. I mean Alice and Bob are both busy people, there's no need to pick out a gift for every single occasion. And I mean Bob gets it, Alice is pretty occupied with work these days. Lately Bob has been trying to clean out his place anyway, so he probably wouldn't have appreciated the extra clutter. It's all cool, is what I'm trying to say here.

[1] Multiple definitions apply here.

[2] Referred to in the context of gift transactions as the "It's the thought that counts" standard.

[3] The "belated" flag on line 3 is an optional header to denote a deferred message. It's inclusion is not necessary in general for empty payloads.

2 Problem Statement

Anyway, this procedure works correctly for well-intentioned senders under the assumption that they did in fact put in the implicit amount of thought required for the sentiment to count. However, this protocol is open to exploit against adversaries who try to send unthoughtful metadata across multiple recipients. Consider the messages A and B in Figures 2 and 3 respectively.

Figure 2: Message A

```
Hey Alice,

Happy birthday! How have you been? Sorry
I couldn't get you something special
this year, but it's the thought that
counts! I'm sure it's been a busy year
for you, with all the recent promotions
and the big merger coming up.


Anyway, hang in there, I know you're
doing your best!

Yours truly,
Eve
```

Associated Payload: $100 Best Buy Gift Card

Figure 3: Message B

```
Hey Bob,

Happy birthday! How have you been? Sorry
I couldn't get you something special
this year, but it's the thought that
counts! I'm sure it's been a busy year
for you, with all the investors backing
out of your impractical blockchain
idea and the divorce with Carol.

Anyway, hang in there, I know you're
doing your best!

Yours truly,
Eve
```

Associated Payload: $20 Starbucks Gift Card[4]

Upon close examination, it's clear that Eve didn't put a complete thought into each message but instead used a common template for both messages effectively achieving a double-spend of thought[5]. Formally, she puts an average of $(1 + \epsilon)/2$ units of thought into each message where the denominator represents the initial one unit of thought[6] put into generating the template and an additional $0 \leq \epsilon < 1$ of thought to adapt the template to a new recipient.

More generally we can say that for n recipients involved in a given template attack, each recipient receives an average $\frac{1+(n-1)\epsilon}{n}$ units of thought. This works out to a real valued amount of thought θ bound by the inequality $0 < \frac{1}{n} \leq \theta \leq \frac{1}{n} + \frac{n-1}{n}\epsilon < 1$. Because this range of values is continuous in the real numbers its cardinality is beyond countably infinite and therefore cannot possibly count.

To further complicate matters, Alice's birthday comes significantly earlier in the year, so Bob is definitely getting a modified version of Alice's card. And I mean really, why did Eve choose to bring those things up? That's just so inconsiderate. Whatever, I'm not going to dwell on it.

[4] $6.54 balance remaining at time of receipt.

[5] Doublethink, for short.

[6] Which, let's be honest, probably wasn't even that much.

Table 1: Header format

Header	Size	Purpose
Hash of the previous blocks header	32B	Prevents precomputation of future blocks before completion of pending block
Hash of the current block's metadata	32B	Prevents precomputation of headers for arbitrary messages
Nonce	32B	Incremented (starting from zero, and carrying over starting position between blocks) to vary the header's hash
Timestamp	8B	Unix timestamp of expected time of receipt (unsigned 64-bit integer)
Difficulty	1B	Determines the range of values within which the hash must fall (unsigned 8-bit integer)

3 Proposal

We propose the Blockcard protocol to mitigate this attack vector. Blockcard is implemented as a series of blocks made up of a set of fixed length headers followed by a variable length record where users can place their metadata. The formats of the headers are shown in Table 1. Blocks are appended in an alternating fashion such that computation cannot begin until the sender receives the latest block on the day of their special occasion.

3.1 Hashes

The hash used in the first two fields are the same double SHA-256 typically used in the popular Bitcoin protocol. The hash of the previous blocks header allows us to chain together blocks in such a way that it is infeasible to censor old messages[7] without forcing the start of a new chain or breaking SHA-256. Since SHA-256 is thought to be a secure one-way function, template attacks like the one shown in Figures 2 and 3 can no longer be used to efficiently perform doublethinks as a new proof-of-thought will have to be completed for each message. New blocks can be checked efficiently by computing the header hash and verifying against the difficulty constraints.

3.2 Nonce

The nonce differs from existing protocols by requiring that iteration begins at the same position as the final value from the previous block. This allows the nonce to serve as a counter to indicate the cumulative amount of thought that has been invested into a chain of transactions. In order to take advantage of parallel computation it is recommended that that individual workers are either passed their nonce value from a main thread that manages the current state of the nonce, or that each thread maintain its own offset copy of the nonce and increase by the total number of workers on each iteration.

3.3 Timestamp

The timestamp field serves as a weak identifier of the recipient and reminder of block computation deadline. The probability of zero collisions falls below 50% for groups with more than 23 individuals[2] but is sufficient for common use where networks tend to be sparse (≈ 4.1 close contacts per person) [3].

3.4 Difficulty of Next Block

Difficulty is used to determine the bounds for the hash value. We differ from other block chain systems by enforcing that the difficulty be equivalent to the age of the recipient at the expected time of message receipt. The difficulty d constrains the value of the hash to be in the range $[2^{255-d}, 2^{255-d+1})$ when interpreted as a big-endian 256-bit integer. This forces the number of leading zeros in the base 2 representation to be exactly the recipient's age. This implicitly acknowledges the recipients age in a semi-obfuscated format, in case it happens to be a sensitive topic, and has the advantage of naturally increasing in difficulty over time. On average, the sender will be required to compute 2^{d+1} hashes to find a valid header.

[7] Knowledge of this property may help to deter adversaries from sending insensitive metadata (see Figure 3) or at the very least incentivize them to take the time to be a little more considerate.

4 Implementation

A working implementation of the protocol can be found at https://github.com/ckw017/blockcard.

5 Future Work

Further research has begun into related methodologies to secure seasonal thank you notes and get-well-soon cards.

References

[1] C. Pignataro. The null packet. RFC 6592, RFC Editor, April 2012.

[2] David Wagner. A generalized birthday problem. In *Annual International Cryptology Conference*, pages 288–304. Springer, 2002.

[3] Pádraig Mac Carron, Kimmo Kaski, and Robin Dunbar. Calling dunbar's numbers. *Social Networks*, 47:151–155, 2016.