Deploying PayWord on Ethereum

***Abstract*—We revisit the 1997 PayWord credit-based micropayment scheme from Rivest and Shamir. We observe that smart contracts can be used to augment ...**

***Keywords*-Cryptocurrency; Bitcoin; Blockchain; Ethereum; Smart Contracts; Payment Channels**

I. Introduction

Beginning in the 1980s, a significant amount of the cryptographic literature has been devoted to the design of e-cash systems. In the 1990s, many startups worked toward deployment of this technology but most ultimately failed [cite textbook]. By late 2008, when Bitcoin was first proposed, innovation on both the academic and commercial side of digital cash had dried up. Now Bitcoin's success has breathed new life into the field: cryptocurrencies have billion dollar market capitalizations and academic conferences like *Financial Cryptography* are again publishing papers on financial cryptography.

At first glance, Bitcoin seems like a major departure from the e-cash systems on the 80s and 90s. In reality, it's 'academic pedigree' is built from existing ideas [c]. Researchers are re-discovering long lost ideas from the e-cash literature and finding new ways to apply them in a blockchain world. For example, blinded coins were a staple of e-cash [c] that re-emerge in systems like zcash [c,c]. Enabling micropayments through lottery-based probablistic payments of macropayments was explored in the 90s [c] and re-emerged for Bitcoin [c]. In this paper, we 're-discover' the 1997 payment system PayWord from Rivest and Shamir [c].

PayWord is a credit-based payment system, envisioned for small payments. The mechanics we will turn to later, but for now, the reader can think of tokens being issued that have some value. The key limitation of PayWord is that tokens do not have inherent value; their value is based on the trust assumption that a counter-party will honour the value ascribed to them. With Ethereum and other blockchain technologies, we can fix this issue by stapling cryptocurrency to the token through the use of a smart contract.

This transformation turns PayWord from a trust-based credit system to a escrow-based payment system; not unlike offline payment channels and networks being proposed for Bitcoin — the Lightning Network being the most prominent [c]. After presenting our system, EthWord, we discuss its relation to payment channels. It is known that an Ethereum-based payment channel will be less complex than a Bitcoin one, since most of the complexity of Bitcoin-based payments channels (*e.g.,* a [c]) comes from Bitcoin's limited scripting language. EthWord a uni-directional (monotonic) payment channel that can be chained into a payment network and has very compact (*e.g.,* 112-bit) payments. It thus might be an interesting primitive to enhance in the same ways other payment channels have been: adding duplex / bi-directionality [c,c], increasing efficiency [c,], and adding transactional privacy[c,].

II. PayWord Primer

PayWord relies on hash chains. A hash chain is constructed by iteratively applying a public one-way hash function $h(.)$ on a random seed value s. To create a hash chain of length N, one first selects the seed value s, then repeatedly applies $h(.)$ on s for N times, thus resulting in a sequence of N hashes:

$$h^N(s), h^{N-1}(s), h^{N-2}(s), \cdots, h^2(s), h(s), s$$

where $h^N(s)$ is called the tip of the chain and can be considered as the public key in a public key cryptosystem, and s is also referred to by $h^0(s)$. More precisely, because of the preimage resistance of one way hash functions, knowing $h^N(s)$ does not reveal the value of s or any of the previous hashes $h^{N-i}(s)$, $i = 1, 2, \cdots, N-1$. On the other hand, by knowing s or any intermediate hash, one can easily verify the correctness of any successive hash values.

In Payword micropayment scheme, a hash chain is used as follows; the tip $h^N(s)$ is first committed by a customer by digitally signing it such that a merchant can verify this signature. Then, for each successive payment, the customer release a previous hash from the chain which can be verified by the merchant that it hashes to $h^N(s)$. Accordingly, a merchant can aggregate such individual i payments and cash one equivalent larger payment by releasing $h^{N-i}(s)$ to the merchant's financial authority which can verify if it hashes iteratively for i times to $h^N(s)$. The customer may use up to N equivalent payments until the seed value s is used. At this time the hash chain is said to be exhausted and the whole process should be re-initialized again with a different seed value.

III. EthWord

EthWord is an Ethereum-based smart contract that enables the aggregation of an offchain set of small payments to one or more onchain larger transactions. Particularly, our contract implements a smart contract moderated version

of the Payword micropayment protocol and mediates the interaction between a customer C and a merchant M. A pseudocode for the proposed smart contract is shown in Figure **??**.

EthWord is designed with two main functions that mediate the transaction process between C and M. First, the contract is instantiated by C through calling the `open` function. In this function, C initializes the contract through passing the pseudonym of the merchant's account M, the tip of the hash chain $\_root$ so that the contract can verify released intermediate hash values against it, the amount that each released hash value is worth $\_wordvalue$, and the total amount of the whole hash chain $\_balance$, and T_{end} which is the time after which remaining funds in the contact's balance are refunded to the customer's account. After calling the `open` function, both the contract and merchant can evaluate the length of the chain and, thus, M can offline know how many intermediate hash values she can accept from C without the need to constantly monitor the state of the contract. Also, M knows that she has to claim aggregated payments before T_{end} or else, M can refund the contract's available balance.

A. Claiming aggregated payments

For each payment in the form of an intermediate hash $x = h^i(s)$, $i = 1, 2, \cdots, N-1$ that M receives from C, she can verify its validity offline by evaluating if $h^{(N-i)}(x) = h^N(s)$ or not. Accordingly, M can decide whether to accept the transaction or decline it in the event of a failed verification. After a few micropayments, where M has acquired a set of intermediate hashes, she invokes the `claim` function in the smart contract by passing to it the last received hash value $wordScratch$, and its order within the hash chain n. Consequently, the contract verifies the validity and the required balance of the supplied inputs against the stored hash chain parameters, and upon successful verification, the account of M is credited by the resulting balance.

B. Payment networks

C. Efficiency

D. Security

EthWord smart contract provides the following guaranteed security properties:

- Fair exchange of services:
- Conducting irrefutable transactions without blockchain monitoring:

IV. Relation to Payment Channels

V. Discussion

References

[1] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun. Evaluating User Privacy in Bitcoin. In *Financial Cryptography*, 2013.

[2] K. J. Arrow, R. Forsythe, M. Gorham, R. Hahn, R. Hanson, J. O. Ledyard, S. Levmore, R. Litan, P. Milgrom, F. D. Nelson, G. R. Neumann, M. Ottaviani, T. C. Schelling, R. J. Shiller, V. L. Smith, E. Snowberg, C. R. Sunstein, P. C. Tetlock, P. E. Tetlock, H. R. Varian, J. Wolfers, and E. Zitzewitz. The promise of prediction markets. *Science*, 320(5878), 2008.

[3] T. Aura, P. Nikander, and J. Leiwo. DoS-resistant authentication with client puzzles. In *Security Protocols*, 2000.

[4] A. Back. Hashcash: a denial of service counter-measure, 2002.

[5] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. Kroll, and E. Felten. Bitcoin and second-generation cryptocurrencies. In *IEEE Symposium on Security and Privacy*, 2015.

[6] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten. Mixcoin: Anonymity for bitcoin with accountable mixes. In *Financial Cryptography*, 2014.

[7] J. Brown. Betting Markets with Decentralized Resolution and Persistent Reputation using Electronic Cash. private communication, Dec. 2013.

[8] D. Chaum. Blind signatures for untraceable payments. In *CRYPTO*, 1982.

[9] Y. Chen and D. M. Pennock. Designing markets for prediction. *AI Magazine*, 2010.

[10] J. Clark, J. Bonneau, A. Miller, J. A. Kroll, E. W. Felten, and A. Narayanan. On decentralizing prediction markets and order books. In *WEIS*, 2014.

[11] J. Clark and A. Essex. Commitcoin: Carbon dating commitments with bitcoin. In *Financial Cryptography*, 2012.

[12] G. G. Dagher, B. Buenz, J. Bonneau, J. Clark, and D. Boneh. Provisions: Privacy-preserving proofs of solvency for Bitcoin exchanges. In *CCS '15: Proceedings of the 22nd ACM Conference on Computer and Communications Security*, October 2015.

[13] G. Di Crescenzo. Privacy for the stock market. In *Financial Cryptography*, 2001.

[14] R. Dingledine, A. Serjantov, and P. Syverson. Blending different latency traffic with alpha-mixing. In *Privacy Enhancing Technologies*, pages 245–257. Springer, 2006.

[15] C. Dwork and M. Naor. Pricing via processing or combatting junk mail. In *CRYPTO*, 1992.

[16] S. Eskandari, D. Barrera, E. Stobert, and J. Clark. A first look at the usability of Bitcoin key management. In *USEC*, 2015.

[17] E. Gabber, M. Jakobsson, Y. Matias, and A. Mayer. Curbing junk e-mail via secure classification. In *Financial Cryptography*, 1998.

[18] S. Haber and W. S. Stornetta. How to time-stamp a digital document. In *CRYPTO*, 1990.

[19] S. A. Haber and W. S. Stornetta. Secure names for bit-strings. In *CCS*, 1997.

[20] R. Hanson. Combinatorial information market design. *Information Systems Frontiers*, 5(1), 2003.

[21] L. Harris. *Trading and exchanges: market microstructure for practitioners*. Oxford, 2003.

[22] M. Jakobsson and A. Juels. Proofs of work and bread pudding protocols. In *Communications and Multimedia Security*, 1999.

[23] A. Juels and J. Brainard. Client puzzles: A cryptographic defense against connection depletion attacks. In *NDSS*, 1999.

[24] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage. A fistful of bitcoins: characterizing payments among men with no names. In *IMC*, 2013.

[25] I. Miers, C. Garman, M. Green, and A. D. Rubin. Zerocoin: Anonymous Distributed E-Cash from Bitcoin. In *IEEE Symposium on Security and Privacy*, 2013.

[26] S. Nakamoto. Bitcoin: A peer-to-peer electionic cash system. Unpublished, 2008.

[27] A. Narayanan, J. Bonneau, E. W. Felten, A. Miller, and S. Goldfeder. *Bitcoin and Cryptocurrency Technologies*. Princeton, 2016.

[28] F. Reid and M. Harrigan. An analysis of anonymity in the bitcoin system. In *Security and Privacy in Social Networks*, 2013.

[29] R. L. Rivest and A. Shamir. PayWord and MicroMint: two simple micropayment schemes. In *Security Protocols*, 1996.

[30] R. L. Rivest, A. Shamir, and D. A. Wagner. Time-lock puzzles and timed-release crypto. Technical Report TR-684, MIT, 1996.

[31] D. Ron and A. Shamir. Quantitative Analysis of the Full Bitcoin

Transaction Graph. In *Financial Cryptography*, 2013.

[32] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *Privacy Enhancing Technologies*, pages 41–53. Springer, 2003.

[33] A. Serjantov, R. Dingledine, and P. Syverson. From a trickle to a flood: Active attacks on several mix types. In *Information Hiding*, pages 36–52. Springer, 2003.

[34] C. Thorpe and D. C. Parkes. Cryptographic securities exchanges. In *Financial Cryptography*, 2007.

[35] C. Thorpe and S. R. Willis. Cryptographic rule-based trading. In *Financial Cryptography*, 2012.

[36] J. Wolfers and E. Zitzewitz. Prediction markets. *Journal of Economic Perspective*, 2004.

[37] J. Wolfers and E. Zitzewitz. Interpreting prediction market prices as probabilities. Technical Report 12200, NBER Working Paper, 2006.

[38] P. F. Yeh. Using prediction markets to enhance us intelligence capabilities: A "standard & poors 500 index" for intelligence. *Studies in Intelligence*, 50(4), 2006.

[39] W. Yuen, P. Syverson, Z. Liu, and C. Thorpe. Intention-disguised algorithmic trading. In *Financial Cryptography*, 2010.