Incentives in Security Protocols

Sarah Azouvi, Alexander Hicks, and Steven J. Murdoch

University College London
{sarah.azouvi.13,alexander.hicks,s.murdoch}@ucl.ac.uk

Abstract. Real world protocols often involve human choices that depend on incentives, including when they fail and require fail-safe or fail-deadly mechanisms. We look at three example systems (the EMV protocol, consensus in cryptocurrencies, and Tor) in this context, paying particular attention to the role that incentives play in fail-safe and fail-deadly situations. We argue that incentives should explicitly be taken into account in the design of security protocols, and discuss general challenges in doing so.

1 Introduction and background

Many real-world systems involve human interaction and decisions that impact the security protocols involved. Protocols can fail, sometimes due to the human side of the system, because of mistakes or malicious behaviour. A way to understand such failures is to look at incentives, as these can determine the human choices involved. Equally, protocol designers can structure incentives to avoid failures of the overall system. Despite the importance of incentives, security proofs very rarely (if ever) explicitly consider the role of incentives as part of the protocols, treating incentives separately.

Incentives come in many shapes and forms. Economic incentives are commonly discussed, following work by Anderson [8], but usually in the context of economic analysis of security problems rather than protocol design. Non-economic incentives also exist, in systems designed to provide security properties like privacy and anonymity that do not involve transactions or handle valuable assets. Differentiating between positive and negative incentives, such as fines or rewards, which serve to discourage or encourage some behaviour can also be useful as they may be perceived differently. Incentives can also be internal (inherent to the protocol) or external (due to factors like legislation) and explicit or implicit if they are derived from other factors, sometimes unexpectedly.

Fail-safe and fail-deadly protocols provide good examples of protocol instances involving incentives, as they handle various types of behaviour. With this in mind, we look at three examples (the EMV protocol, consensus in cryptocurrencies and Tor) before discussing the integration of incentives into protocols.

International Workshop on Security Protocols, Cambridge, UK, March 2018 (pre-proceedings draft).

2 Incentives in existing systems

2.1 EMV

The EMV protocol is used for the vast majority of smart card payments worldwide, and also is the basis for both smartphone and card-based contactless payments. Over the past 20 years, it has been gradually refined as vulnerabilities have been identified and removed. However, there is still considerable fraud which results, not from unexpected protocol vulnerabilities, but from deliberate decisions that participants can make to reduce the level of security offered. Such decisions include the bank omitting the cards' ability to produce digital signatures (making cards cheaper but easy to clone), the merchant omitting PIN verification (making transactions faster, but stolen cards easier to use), or the payment network not sending transaction details back to the bank that issued the card for authorisation when the card is used abroad (reducing transaction latency, but making fraud harder to prevent).

Fraud exploiting such decisions is not strictly speaking a protocol failure but, if unchecked, could be financially devastating for participants and reduce trust in the system. The way in which the payment industry has managed the risk is through incentives: firstly, reducing fees for transactions that use more secure methods, and secondly, assigning the liability for fraud to the party which causes the security level to be reduced. Any disputes are handled as specified by the relevant contracts, whether in court or through arbitration.

Looking at the EMV ecosystem as a whole, this serves as a fail-safe overlay on top of a protocol which is optimised for compatibility rather than security. While any individual transaction could go wrong, over time, parties will be encouraged to either adopt more secure options or mitigate fraud in other ways, for example through machine-learning based risk analysis. However, there is little indication that the EMV protocol was designed with the understanding that incentives would play such a central role in the security of the system.

Where this omission becomes particularly apparent is that during disputes, it may be unclear how a fraud actually happened, leading to a disagreement as to who should be liable. This is because communication between participants is designed to establish whether the transaction should proceed, rather than which party made which decision. Importantly, the policies on how participating entities should act are not part of the EMV specification. Even assuming that all participants in a dispute are acting honestly, it can be challenging for experts to reverse-engineer decisions from the limited details available [29].

This suggests that where incentives are part of the fail-safe mechanism, the protocol should produce unambiguous evidence showing not only the final system state, but how it was arrived at. This evidence should also be robust to participants acting dishonestly, perhaps through use of techniques inspired by distributed ledgers [30]. Currently only a small proportion of the protocol exchange has end-to-end security, but because payment communication flows are only between participants with a written contract (for historical, rather than technical reasons) this deficiency is somewhat mitigated. We know how to rea-

son about the security of protocols, but what would be an appropriate formalisation that would indicate whether evidence produced by a system is sufficient to properly allocate incentives?

2.2 Consensus protocols in cryptocurrencies

We move on to consider cryptocurrencies (e.g. Bitcoin, Ethereum), public distributed ledgers relying on a blockchain and consensus protocol. Originating from the rejection of any centralised authority, these are a rare example of systems whose security inherently relies on incentive schemes, unlike the EMV protocol above. Transactions are verified and appended to the blockchain by miners incentivised by mining rewards and transaction fees defined in the protocol to encourage honest behaviour in a trustless, open system.

This has had notable success, but does not address every possible issue as attacks on Bitcoin mining exist [11, 13, 16, 17, 28, 34], suggesting that the incentives defined in the Nakamoto consensus protocol do not capture all possible behaviours. Despite all these attack papers discussing incentives, few other papers focus on them [4, 26, 27, 31], and security oriented papers consider them separately [10, 25, 32].

These also focus on standard game theoretic concepts like Nash equilibria [10, 25, 31] and assume rational participants, whilst distributed systems aim for security properties like Byzantine Fault Tolerance to tolerate a subset of participants arbitrarily deviating. (This is with the exception of recent work by Badertscher et al. [9] that considers mining in the setting of rational protocol design.) Some attacks are also not appropriately studied from the point of view of Nash equilibria, as they are often on the network layer of the protocol, as in the case of selfish mining [17]. The fact that papers considering incentives tackle these attacks separately also points to the fact that Nash equilibria are not well suited for this context. Indeed, there is a mismatch between a Nash equilibria, which exist in the context of finite action games involving a finite set of participants, and systems such as consensus protocols where the set of actions is theoretically unlimited as one could try to build alternative chains or broadcast their blocks at any time.

Examples of incentive based fail-safe and fail-deadly instances of the consensus protocol can be found in forking mechanisms, which can be used to incorporate new rules or revert to a previous state of the blockchain. Soft forks are an example of a fail-safe mechanism as even in the case of a disagreement amongst network peers, they are backwards compatible and allow peers to choose what software to run without splitting the network. When no such compatibility can be found, the network can implement a hard fork where every peer has to comply with the new rules. On the other hand, if a hard fork is implemented without the consent of the whole network, it may split like Ethereum after the DAO hack [35]. Due to part of the network having clear incentives to roll back, a hard fork was organised to reverse the state of the blockchain to a moment before the hack. This caused controversy, as some considered it to go against the ideology

of decentralisation, causing part of the network to split and create a new currency, Ethereum Classic [1], in which the hack remained. Nonetheless, forking and splitting up the network could lessen its utility (Ethereum Classic is now worth much less than Ethereum [14]), which gives a fail-deadly case since miners would risk losing mining rewards and the cost of creating a block if their fork is not supported.

Finally, there is the case of protocols added on top of the system such as the Lightning off-chain payment channel system [33] which allows two or more parties to transact offline, publishing only two on-chain transactions: a deposit which locks funds and a final balance which settles the payment. Although this involves cryptography, the security is largely based on incentives: parties are disincentivised from cheating (by publishing an old transaction to the blockchain) as the honest party could then broadcast a revocation transaction (signed by the cheating party) and receive the deposit of the cheating party. This fail-deadly case is not unlike the EMV protocol case, where robust evidence may deter dishonest behaviour.

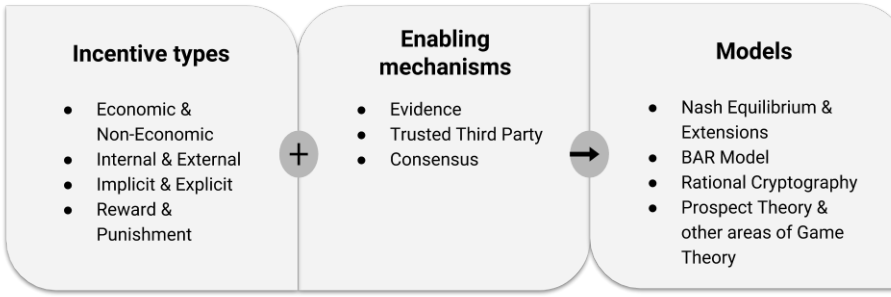
2.3 Incentives in non-economic systems

The above examples illustrate systems involving transactions, but what of systems which do not involve transactions or valuable assets? We consider this case by looking at the anonymity system Tor, whose security relies on the number of participants and servers in the network. Whilst there may be incentives for many (perhaps not all) users of the Tor network, there is less incentive to host a Tor server. Nevertheless, the network has grown to around 4 million users and 6 thousand servers as of January 2018.

Clearly, the lack of economic incentives does not prevent the existence of Tor but perhaps they could motivate users to participate and host servers. The economics of anonymity have been studied, dating back to at least the early noughties [6] and proposals to reward hosting servers have been made [19,23] but not implemented. Performance based incentives were also considered by Ngan et al. [15]. Incentives to avoid security failures like sending traffic through a bad node could also be considered, as robust evidence of a nodes status would provide a fail-safe (participants could avoid sending traffic through it) and fail-deadly (by punishing the host) mechanism.

But whilst adding incentives may improve the performance and security of the network, it may also produce unexpected results. A relevant study is the work of Gneezy and Rustichini [20], who looked at the effects of implementing incentives (fines in their case) to late parents at a nursery. This resulted in parents coming even later, a change which was not reverted once fines were removed. They concluded that adding incentives to a system could irreversibly damage it. Simulating the reaction of network participants is very challenging (compared to network performance [22]), which is likely the reason we have seen little experimentation around incentives.

Fig. 1. Summary incentive types, enabling mechanisms and models



3 Discussion and conclusion

The previous section serves to illustrate the role incentives can play in the security of a system. From what we've discussed, there are three important aspects to consider: incentive types, mechanisms that enable them and models to reason about them.

Incentive types are divided into economic or non-economic, external and internal, explicit and implicit, and rewards and punishments (see Figure 1). For most real world examples, economic incentives may seem like a natural choice if an exchange of valuable services, goods or currency but that is not always the case. However, non-economic incentives can also be required but it is much less clear how their utility can be evaluated, especially by the parties meant to be enticed. To evaluate the utility of an incentive, it should also be explicit. Implicit incentives are more likely to end up being exploited, as described in many of the attacks on mining. These are also linked to internal incentives, which are easier to abuse, rather than external incentives which might require convincing an external party to collude. Thus the type of incentive might have an impact not only on the utility derived from incentives, but also on the security of the system if they are more likely to be exploited. Rewards and punishments are also to be considered, to incentivise honest behaviour, or disincentivise dishonest behaviour, depending on which is costlier or applicable to the context.

In order for incentives to work, they must be reliable in the sense that any party can expect (or rather, be guaranteed) to receive the related payoff. For all of the examples we considered, evidence is used by parties to ensure an incentive's payoff can be obtained. It is natural to expect evidence would be

required, decisions are made based on information and as payoffs are enforced by external parties (e.g. the justice system in the EMV case, or the network in cryptocurrencies) which should not reward or punish anyone without verifiable evidence. Nonetheless, it would be interesting to determine if other mechanisms could be used in place of, or on top of evidence to make incentives reliable and ensure agents in the system do not ignore them.

Once a type and enabling mechanism is chosen, it is necessary to have a framework that allows us to reason about them. The main challenge is obtaining a framework that allows reasoning about incentives on a level similar to security protocols. Standard game theoretic concepts like Nash equilibria, which only consider up to one participant deviating, are not enough when dealing with distributed systems that tolerate far more, as well as information asymmetry, asynchronicity and cost of actions. Such issues are discussed by Halpern [21], who provides an overview of extensions of the Nash equilibrium. Appendix A provides informal definitions for these concepts (as well as a few others). For example, (k,t)-robustness combines k-resilience (tolerating k participants deviating) and t-immunity (participants who do not deviate are not worse off for up to t participants deviating). Introduced by Abraham et al. [2], in the context of secret sharing and multiparty computation, this better fits fail-safe guarantees (e.g. Byzantine Fault Tolerance) we expect from systems. Solidus [4] uses this concept to provide an incentive-compatible consensus protocol, although they address selfish mining separately from the rest of the protocol.[1] We've also considered fail-deadly cases, which are usually addressed through deterrence. A good fit for these are (k,t)-punishment strategies, where the threat of t participants enforcing a punishment stops a coalition of k participants from deviating. These definitions are only a start to bridging the gap between Game Theory and Computer Security settings. Other work in that direction includes the BAR model [7], which combines Game Theory and Distributed Systems and considers three types of participants (Byzantine, Altruistic and Rational) and the field of Rational Cryptography [9,12,18] which combines Game Theory and Cryptography by using cryptographic models with rational agents.

Although the above does not capture all we could want from a system, we may now wonder what a security proof involving incentives would look like. In many ways, the current standard of security proofs involves games and probabilistic arguments. This is not far removed from game theoretic proofs concerned with strategies (especially in incomplete information games), although it requires bridging the differences in settings explored in the last paragraph. Evaluating the assumptions underlying incentives, and not only their impact, would be necessary. For example in the EMV protocol and Lightning network, both rely on evidence generated by the protocols for their fail-deadly uses. Incentives would also have to be weighted by the robustness of the mechanisms they relate to. For example, evidence based deterrence in fail-deadly instances is only as good as the evidence generated. Whilst proving robustness of the evidence is realistic for

[1] Although included in the first version of the pre-print, the published version [5] states that rigorous analysis of incentives is left to future work.

cryptographic evidence, legislation or other factors (social, moral, economic) are much harder to formally evaluate (although Prospect Theory [24] may provide some tools) even if assumptions about altruistic behaviour can clearly be made in cases such as Tor.

References

1. Ethereum Classic. https://ethereumclassic.github.io/.
2. I. Abraham, D. Dolev, R. Gonen, and J. Halpern. Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Principles of Distributed Computing*, PODC '06, pages 53–62, New York, NY, USA, 2006. ACM.
3. I. Abraham, D. Dolev, and J. Y. Halpern. Lower bounds on implementing robust and resilient mediators. In *Theory of Cryptography Conference*, pages 302–319. Springer, 2008.
4. I. Abraham, D. Malkhi, K. Nayak, L. Ren, and A. Spiegelman. Solidus: An incentive-compatible cryptocurrency based on permissionless Byzantine consensus. *CoRR*, abs/1612.02916, 2016.
5. I. Abraham, D. Malkhi, K. Nayak, L. Ren, and A. Spiegelman. Solida: A blockchain protocol based on reconfigurable Byzantine consensus. In *OPODIS*, 2017. https://eprint.iacr.org/2017/1118.
6. A. Acquisti, R. Dingledine, and P. Syverson. On the economics of anonymity. In *Financial Cryptography and Data Security*, pages 84–102. Springer, 2003.
7. A. S. Aiyer, L. Alvisi, A. Clement, M. Dahlin, J.-P. Martin, and C. Porth. BAR Fault Tolerance for cooperative services. *SIGOPS Oper. Syst. Rev.*, 39(5):45–58, Oct. 2005.
8. R. Anderson. Why information security is hard – an economic perspective. In *Annual Computer Security Applications Conference*, pages 358–365, 2001.
9. C. Badertscher, J. Garay, U. Maurer, D. Tschudi, and V. Zikas. But why does it work? a rational protocol design treatment of bitcoin. Technical report, Cryptology ePrint Archive, Report 2018/138, 2018. https://eprint. iacr. org/2018/138, 2018.
10. I. Bentov, R. Pass, and E. Shi. Snow White: Provably secure proofs of stake. *IACR Cryptology ePrint Archive*, 2016:919, 2016.
11. J. Bonneau. Why buy when you can rent? In *Financial Cryptography and Data Security*, pages 19–26. Springer, 2016.
12. P. Caballero-Gil, C. Hernández-Goya, and C. Bruno-Castañeda. A rational approach to cryptographic protocols. *CoRR*, abs/1005.0082, 2010.
13. M. Carlsten, H. Kalodner, S. M. Weinberg, and A. Narayanan. On the instability of Bitcoin without the block reward. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 154–167, New York, NY, USA, 2016. ACM.
14. CoinMarketCap. Cryptocurrency market capitalizations. https://coinmarketcap.com/. Accessed: 2018-01-15.
15. R. Dingledine, D. S. Wallach, et al. Building incentives into Tor. In *Financial Cryptography and Data Security*, pages 238–256. Springer, 2010.
16. I. Eyal. The miner's dilemma. In *IEEE Symposium on Security and Privacy*, 2015.
17. I. Eyal and E. G. Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *Financial Cryptography and Data Security*, 2013.

18. J. Garay, J. Katz, U. Maurer, B. Tackmann, and V. Zikas. Rational protocol design: Cryptography against incentive-driven adversaries. Cryptology ePrint Archive, Report 2013/496, 2013. http://eprint.iacr.org/2013/496.
19. M. Ghosh, M. Richardson, B. Ford, and R. Jansen. A TorPath to TorCoin: Proof of Bandwidth altcoins for compensating relays. Technical report, NRL, 2014.
20. U. Gneezy and A. Rustichini. A fine is a price. *The Journal of Legal Studies*, 29(1):1–17, 2000.
21. J. Y. Halpern. Beyond Nash equilibrium: Solution concepts for the 21st century. *CoRR*, abs/0806.2139, 2008.
22. R. Jansen and N. Hopper. Shadow: Running Tor in a box for accurate and efficient experimentation. In *Proceedings of the 19th Symposium on Network and Distributed System Security (NDSS)*. Internet Society, February 2012.
23. R. Jansen, A. Miller, P. Syverson, and B. Ford. From onions to shallots: Rewarding Tor relays with TEARS. Technical report, NRL, 2014.
24. D. Kahneman and A. Tversky. Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I*, pages 99–127. World Scientific, 2013.
25. A. Kiayias, A. Russell, B. David, and R. Oliynykov. Ouroboros: A provably secure Proof of stake blockchain protocol. In *Annual International Cryptology Conference*, pages 357–388. Springer, 2017.
26. A. Kothapalli, A. Miller, and N. Borisov. SmartCast: An incentive compatible consensus protocol using smart contracts. In *1st Workshop on Trusted Smart Contracts, Financial Cryptography*, 2017.
27. J. A. Kroll, I. C. Davey, and E. W. Felten. The economics of Bitcoin mining, or Bitcoin in the presence of adversaries. In *WEIS*, 2013.
28. L. Luu, J. Teutsch, R. Kulkarni, and P. Saxena. Demystifying incentives in the consensus computer. In *Computer and Communications Security*, CCS '15, pages 706–719, New York, NY, USA, 2015. ACM.
29. S. J. Murdoch. Reliability of Chip & PIN evidence in banking disputes. *Digital Evidence and Electronic Signature Law Review*, 6, 2009.
30. S. J. Murdoch and R. Anderson. Security protocols and evidence: Where many payment systems fail. In *Financial Cryptography and Data Security*, pages 21–32. Springer, 2014.
31. S. Park, K. Pietrzak, A. Kwon, J. Alwen, G. Fuchsbauer, and P. Gai. SpaceMint: A cryptocurrency based on proofs of space. Cryptology ePrint Archive, Report 2015/528, 2015. https://eprint.iacr.org/2015/528.
32. R. Pass and E. Shi. Fruitchains: A fair blockchain. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*, pages 315–324. ACM, 2017.
33. J. Poon and T. Dryja. The Bitcoin lightning network: Scalable off-chain instant payments. *Technical Report (draft)*, 2015.
34. A. Sapirshtein, Y. Sompolinsky, and A. Zohar. Optimal selfish mining strategies in bitcoin. In J. Grossklags and B. Preneel, editors, *Financial Cryptography and Data Security*, pages 515–532, Berlin, Heidelberg, 2017. Springer Berlin Heidelberg.
35. D. Siegel. Understanding the DAO attack. https://www.coindesk.com/understanding-dao-hack-journalists/, June 2016.

A Glossary

For completeness, we add a glossary of terms and definitions appearing in the main content of the paper, particularly in the discussion section. Note that we

keep the definition fairly informal so that they can easily be referenced, more formal definitions can be found work referenced in the main sections of the paper [2, 3, 7, 12, 18, 21, 24].

Nash equilibrium: In a game of n players with corresponding strategy sets and payoff functions for each strategies, the strategy profile is the tuple of strategies selected by each player. A strategy profile is a Nash equilibrium for the game if no unilateral deviation by any single player is profitable for that player.
Note that this presumes players have knowledge of the game and possible strategies for all players in the game. Moreover, it is only concerned with single players deviating rather than multiple (independent or colluding) players deviating.

k-resilience: A Nash equilibrium is said to be k-resilient if a coalition of up to k players cannot increase their utilities by deviating, given that the rest of the players do not deviate.

Nash equilibrium: In a game of n players with corresponding strategy sets and payoff functions for each strategies, the strategy profile is the tuple of strategies selected by each player. A strategy profile is a Nash equilibrium for the game if no unilateral deviation by any single player is profitable for that player.

t-immunity: A strategy profile is said to be t-immune if players who do not deviate are no worse off for up to t players deviating.

(k,t)-robustness: A strategy profile is said to be (k, t)-robust if it is both k-resilient and t-immune.
Note that a Nash equilibrium is $(1, 0)$-robust, and for $(k, t) \neq (1, 0)$ there does not generally exist a (k, t)-robust equilibrium. Aside from equilibria, (k, t)-robust strategies do exist in certain games, particularly when a *mediator* can be considered as in the case of a Byzantine agreement where the mediator relays the preference of the general to the soldiers (including t traitors). More generally, a mediator could be implemented through gossiping between players although this depends on the the number of players as well as the parameters (k, t) [21] and can depend on a (k, t)-punishment strategy.

(k,t)-punishment: A (k, t)-punishment strategy is such that if k players deviate, they do not increase their utility as long as t players enforce the punishment.

BAR model: In Distributed Systems, the BAR model was introduced [7] to incorporate rational participants as in game theoretic models. Traditionally, the Distributed Systems literature considers good and bad processes in (for example) crash fault tolerant or Byzantine fault tolerant settings. The BAR model differs by considering player of three types: Byzantine players that act randomly, altruistic players that comply with the protocol and rational players that maximise their expected utility.

Rational Cryptography: The field of Rational Cryptography [12, 18] incorporates incentives within traditional cryptographic systems. They consider each party in the protocol as an agent trying to increase their expected utility. Rational Protocol Design is another variant of this, introduced by Garay

et Al. [18], that considers a cryptographic protocol as a zero-sum game between the protocol designer and the adversary.

Forks and chain splits: In a blockchain based distributed ledger, forking can happen in two situations. Either when two blocks are found by different miners at the same time or in the case of a change in the protocol. Note that this definition is different that the one traditionally used in open source systems. *Soft-forks* are backward compatible changes, meaning that if a node in the system decides to not update their software and stay with the unchanged protocol, their blocks will still be accepted by other nodes. On the other hand a *hard fork* is not backward compatible, meaning that every node in the system needs to run the updated software following the fork. In the case where a subset of nodes decide to not update their software, a *chain-split* may occur, resulting in a split network.

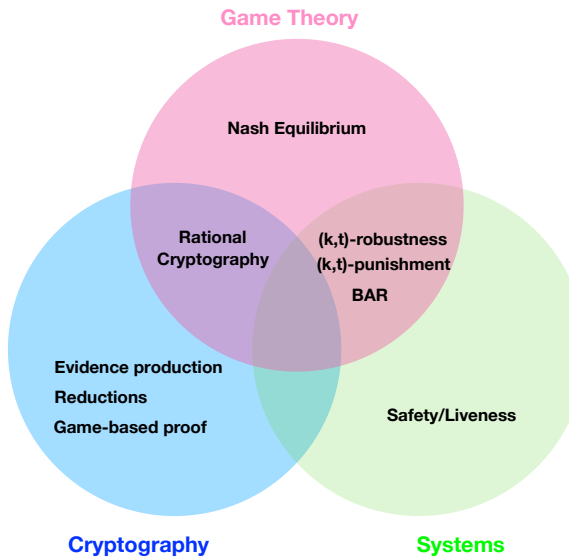


Fig. 2. Different models by field