Why a Digital Euro should be Online-first and Bearer-based

Christian Grothoff Florian Dold

March 29, 2021

The European Central Bank's "Report on a Digital Euro" [ecb20] considers two distinct types of designs for a digital euro. It argues that all functional requirements laid out in the report can be fulfilled by operating the two systems in parallel:

1. A bearer-based digital euro based on trusted hardware that can be used offline, anonymously, and without third-party intervention.

2. An account-based digital euro that can be used online, is fully software-based and excludes the possibility of anonymity.

The report does not discuss other choices of hybrid systems. However, the choice is more arbitrary than it might seem at first sight: bearer-based systems are not necessarily offline payment systems, and online payment systems do not need to exclude anonymity.

We argue that operating a bearer-based payment system to complement an account-based CBDC in order to gain offline and privacy features is not a good trade-off. Adding permanent, regular offline capabilities via the bearer-based payment instrument constantly exposes the CBDC to the severe issues inherent in offline-capable payment systems. Instead, the offline mode of operation should be restricted to scenarios where it is actually required, which mitigates the risks.

1 Challenges of offline payments

Payment processing involves a distributed, networked system. Three properties are desirable for distributed systems:

- Consistency: there is one coherent view of the state of the system and no contradictory believes held by different components.

- Availability: the system is "always" able to provide its service, which implies making updates to the state to perform transactions for the system's users.

- Partition tolerance: the system tolerates network or component failures which makes communication between parts of the distributed system temporarily impossible.

The well-known CAP theorem [GL02] proves that it is impossible to design a network protocol that simultaneously achieves all three properties. For electronic payment systems, this means it is impossible to simultaneously protect against double-spending (Consistency) while operating (Available) offline (Partition-tolerance). Thus, any offline electronic payment system is left with one of the following choices:

- Protect against double-spending by taking away control over computing from the user, typically using hardware security elements that prevent the user from accessing certain functions of the device.[1]
- Retroactively identifying the user after network connectivity is restored, in privacy-preserving systems using conditional deanonymization, and attempting to recoup the losses from the double-spending party afterwards.[2]

There is no third choice. While there are minor variations how one could implement these designs (like blaming the merchant and forcing merchants to cover the double-spending cost), the list is basically exhaustive.

1.1 Hurting security

If breaking the restrictive computing element's security properties gives users the ability to access virtually unlimited funds, they will. Hardware protections typically fall against well-equipped adversaries with plenty of time and expertise.[3] When Google published an attack on ARMs TrustZone, a key observation of the report (that is not uncommon for these types attacks) is:

> "Unfortunately, the design issue outlined in this blog post is difficult to address, and at times cannot be fixed without introducing additional dedicated hardware or performing operations that risk rendering devices unusable." – `https://googleprojectzero.blogspot.com/2017/07/trust-issues-exploiting-trustzone-tees.html`

So hardware security is hardly in a better shape than software security, and issues can be significantly more expensive to fix.

Given a known vulnerability in an offline payment system, nation-state attackers and organized crime may even find it advantageous to force large-scale network outages to bring the payment system into a stage where they can multi-spend.

[1] A good example for such a design is [CGK$^+$20].

[2] A classical example for such a design is [CFN90].

[3] Examples of vulnerabilities in such hardware security systems include [DKK17, GNBD16, LGS$^+$16, MGS$^+$17, TSS17, ZSS$^+$16, Goo20, NBB06, Lak20, Lab19, But20, LZLS19], affecting all major hardware security architectures (Intel, Samsung, ARM, AMD, and SIM cards).

Deanonymization is similarly problematic, as the identified individual may actually be the victim of a computer crime. Furthermore, even if the guilty party is identified, it is unclear that they would be able to cover the costs of the multi-spend. At scale, the resulting potential attacks could endanger financial stability.

1.2 Hurting informational self-determination

Both of the above choices hurt the user's fundamental human right to informational self-determination. Forcing users to use hardware that they do not control is limiting their ability to control and customize their digital lives.

Even in privacy-friendly systems (like those based on Chaum [CFN88]) where citizens can use digital cash to make purchases anonymously, adding the ability to retroactively deanonymize double-spending users implies that accidentally double-spending (say after restoring from backup) voids the privacy assurances of the system. This key security property of the privacy-friendly systems would thus need to be weakened and becomes brittle.

1.3 Hurting availability

A hardware-based solution not only limits availability to those users that can afford the device, but also limits user's ability to make backups of their digital cash. Thus, loosing the hardware will result in citizens loosing their digital cash, something a software-based solution can avoid.

If a hardware-based solution were to enable users making arbitrary backups of their digital cash, it would have to again include a mechanism to reveal the user's identity if double spending is detected. In this case, the solution would fail to offer good privacy protections.

Regardless of hardware or software solutions for offline payments, all such systems where double-spending is detected and the double-spender is retroactively identified and later penalized, the resulting financial risks will create pressures to deny access to the payment system to citizens with insufficient reputation or credit score.

One argument for offline CBDC is the objective to improve availability in situations where network access is unreliable. However, today network availability is usually only problematic in areas where access to electrical power is similarly limited. Thus, in these cases preserving physical cash will help much more, while an offline CBDC is unlikely to significantly improve availability.

1.4 Hurting innovation

In a world where everything is headed towards software solutions, a mandatory hardware security solution for a CBDC is restrictive, not just for customers but also for businesses who want to process payments or offer services related to payments. Furthermore, to ensure the security of the solution, the production of approved hardware devices would need to be strictly controlled, likely reinforcing existing anti-competitive monopolies in the hardware market.

1.5 Hurting cash

The ability to continue to use physical cash is priced by central banks, citizens and experts in disaster management. In situations with wide-spread and lasting power outages, digital systems fail, and thus having cash as a fall-back is crucial. Children find it easier to learn about money if it is physical and not obscured by an electronic abstraction. Thus, availability and accessibility of physical cash will always be unmatched by electronic solutions. A CBDC that competes with cash by providing offline functionality has a higher potential of harming the use of cash than a CBDC that is online-only.

2 Conclusion

While in some situations, offline payments might be a desirable requirement, adding offline payment systems have inherent and severe risks. The exposure to these risks should be limited by only resorting to an offline fallback mode of the payment system when actually required.

Discouraging the use of the offline fallback mode can be easily achieved by by exposing the payee to counterparty risk. In a system based on restricted hardware elements, the payee would bear the risk in case of a compromised hardware system. In a system based on identifying offline double spenders / cheaters, the payee would bear the risk in case the offline double spender / cheater cannot be found and held accountable.

Preliminary results from a survey done by the ECB have shown that privacy is regarded as one of the the most important feature by participants among citizens and businesses. Only providing privacy in the offline payment instrument would make privacy a second class citizen, especially as privacy is important in innovative online usages of a digital euro.

Thus, our improved suggestion for a secure, robust and privacy-friendly digital euro would be the following hybrid:

1. An online, bearer-based payment instrument with anonymity and income transparency features [CFN88, CGM21]. Note that in this proposal, only one of the two parties (payer, payee) needs to have network connectivity.

2. A limited and optional offline mode for the first payment system. The payee must decide whether to accept an offline payment, based on the counterparty risk. The risk depends on the details of the offline payment implementation (hardware security vs. fraudulent party identification).

3. Physical cash as a fallback for emergency situations where power outages or cyber attacks render a digital euro temporarily unusable.

Acknowledgements

We thank Thomas Moser for insightful comments on an earlier draft of this text.

References

[But20] Peter Buttler. Wib vulnerability: Sim-card that allows hackers to takeover phones. https://readwrite.com/2020/01/06/wib-vulnerability-sim-card-that-allows-hackers-to-takeover-phones/, January 2020.

[CFN88] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In *Conference on the Theory and Application of Cryptography*, pages 319–327. Springer, 1988.

[CFN90] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In Shafi Goldwasser, editor, *Advances in Cryptology — CRYPTO' 88*, pages 319–327, New York, NY, 1990. Springer New York.

[CGK$^+$20] Mihai Christodorescu, Wanyun Catherine Gu, Ranjit Kumaresan, Mohsen Minaei, Mustafa Ozdayi, Benjamin Price, Srinivasan Raghuraman, Muhammad Saad, Cuy Sheffield, Minghua Xu, and Mahdi Zamani. Towards a two-tier hierarchical infrastructure: An offline payment system for central bank digital currencies, 2020.

[CGM21] David Chaum, Christian Grothoff, and Thomas Moser. How to issue a central bank digital currency, 2021.

[DKK17] M. Dorjmyagmar, M. Kim, and H. Kim. Security analysis of samsung knox. In *2017 19th International Conference on Advanced Communication Technology (ICACT)*, pages 550–553, 2017.

[ecb20] Report on a digital euro. https://www.ecb.europa.eu/pub/pdf/other/Report_on_a_digital_euro~4d7268b458.en.pdf, October 2020.

[GL02] Seth Gilbert and Nancy Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *SIGACT News*, 33(2):51–59, June 2002.

[GNBD16] R. Guanciale, H. Nemati, C. Baumann, and M. Dam. Cache storage channels: Alias-driven attacks and verified countermeasures. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 38–55, May 2016.

[Goo20] Dan Goodin. Intel sgx is vulnerable to an unfixable flaw that can steal crypto keys and more. Technical report, ARS Technica, 2020.

[Lab19] Security Research Labs. New sim attacks de-mystified, protection tools now available. https://srlabs.de/bites/sim_attacks_demystified/, 2019.

[Lak20] Ravie Lakshmanan. Intel cpus vulnerable to new 'sgaxe' and 'crosstalk' side-channel attacks. https://thehackernews.com/2020/06/intel-sgaxe-crosstalk-attacks.html, June 2020.

[LGS+16] Moritz Lipp, Daniel Gruss, Raphael Spreitzer, Clémentine Maurice, and Stefan Mangard. Armageddon: Cache attacks on mobile devices. In *Proceedings of the 25th USENIX Conference on Security Symposium*, SEC'16, page 549–564, USA, 2016. USENIX Association.

[LZLS19] Mengyuan Li, Yinqian Zhang, Zhiqiang Lin, and Yan Solihin. Exploiting unprotected i/o operations inamd's secure encrypted virtualization. In *USENIX Security Symposium*, 2019.

[MGS+17] Aravind Machiry, Eric Gustafson, Chad Spensky, Christopher Salls, Nick Stephens, Ruoyu Wang, Antonio Bianchi, Yung Ryn Choe, Christopher Kruegel, and Giovanni Vigna. Boomerang: Exploiting the semantic gap in trusted execution environments. In *NDSS*, 2017.

[NBB06] Alexander Nilsson, Pegah Nikbakht Bideh, and Joakim Brorsson. A survey of published attacks on intel sgx. https://arxiv.org/pdf/2006.13598v1.pdf, 2006.

[TSS17] Adrian Tang, Simha Sethumadhavan, and Salvatore Stolfo. Clkscrew: Exposing the perils of security-oblivious energy management. In *Proceedings of the 26th USENIX Conference on Security Symposium*, SEC'17, page 1057–1074, USA, 2017. USENIX Association.

[ZSS+16] Ning Zhang, Kun Sun, Deborah Shands, Wenjing Lou, and Y Thomas Hou. Truspy: Cache side-channel information leakage from the secure world on arm devices. *IACR Cryptol. ePrint Arch.*, 2016:980, 2016.