Formal Verification of the Kate-Zaverucha-Goldberg Polynomial Commitment Scheme

Tobias Rothmann
School of Computation, Information and Technology
Technical University of Munich, Germany
0009-0000-2343-5793

Katharina Kreuzer
School of Computation, Information and Technology
Technical University of Munich, Germany
0000-0002-4621-734X

***Abstract*—When the Ethereum blockchain got updated to use the Kate-Zaverucha-Goldberg (KZG) polynomial commitment scheme (PCS) in March 2024, a rigorous and formal proof of the KZG becomes necessary for more security. This can be achieved by formalization in an interactive theorem prover like Isabelle, as we will outline in this paper. We give an insight into the formalization of the KZG commitment scheme in Isabelle and discuss the difficulties and advantages. This work outlines — to our knowledge — the first formalization of polynomial commitment schemes so far. It is a vital and foundational step towards more reliably safe and secure blockchains and applications in various other cryptographic protocols.**

***Index Terms*—KZG commitment scheme, security, verification, Isabelle.**

I. Introduction

When the Ethereum blockchain updated its protocols in March 2024, it changed some underlying crypto schemes — now using the polynomial commitment scheme (PCS) by Kate, Zaverucha and Goldberg (KZG). As Ethereum is the blockchain with the largest market capitalization after Bitcoin, it is essential to show the security of the blockchain and its cryptographic components.

In order to validate the security of cryptographic protocols such as blockchains, we first need to prove all cryptographic primitives secure. In our case, we analyse the KZG polynomial commitment scheme (in the following abbreviated just by KZG) for the Ethereum blockchain. A commitment scheme is a primitive where one party commits to a chosen (but hidden) value and the other party may later verify this commitment once it is revealed. For polynomial commitment schemes, the values we commit to are polynomials, also allowing a pointwise revelation in the commitment. The KZG scheme is the first and most widely used PCS.

A trend in cryptographic security proofs is to provide rigorous security specifications and proofs of cryptographic primitives and protocols in formal theorem provers. Such formalizations and verifications may uncover security flaws and improve the trustworthiness in the security of crypto primitives or protocols. A real-world example for such a discovery of security issues is the formalization of the protocols for Jitsi video conferences [15] and Matrix [1].

A. Related work

The KZG [14] was the first PCS and is still the most efficient and widely used PCS. Nowadays, there are several types of PCS, including pairing-based approaches like the KZG (e.g. [6]). Other constructions of PCSs use groups of unknown order [9], reed-solomon codes [2], [4], [5] or inner-product-arguments [7]. In our work, we formalize the KZG using the theorem prover Isabelle [16], [17] using libraries contained in the Archive of Formal Proofs [13]. Other efforts in formalizing cryptography include EasyCrypt [12], CryptoVerif [8] and CryptoLine [11].

B. Contributions

With this paper, we outline an ongoing effort to formally state and verify the specifications and security properties of KZG. Our work is build on the CryptHOL [3] framework in the theorem prover Isabelle [17]. In a first step, we define the KZG, give a formal specification and verify the correctness. In a second step, we formally verify the most important security properties: polynomial binding, evaluation binding and hiding. To our knowledge, this is the first formal verification of a polynomial commitment scheme in a theorem prover so far.

II. Formal Specifications and Correctness of KZG

Let us look at a PCS as a protocol between a committer (Alice) and a verifier (Bob). After a key generation, Alice commits to a polynomial and calculates a commitment for this polynomial. Later, Alice can reveal the polynomial (possibly pointwise), such that Bob can verify the commitment to the polynomial. The KZG is a construction of a PCS.

In the following, let p be a prime and t a natural number that is large in comparison to p. Let $\mathbb{G}_p$ denote the group of the KZG and $\mathbf{g}$ a fixed generator.

The KZG has a trusted setup, namely the key generation denoted by $KeyGen$, which generates the public key PK for the main protocol. $KeyGen$ samples a uniformly random field element $\alpha \in \mathbb{Z}_p$ and outputs the public key $PK = (\mathbf{g}, \mathbf{g}^{\alpha}, \mathbf{g}^{\alpha^2}, \ldots, \mathbf{g}^{\alpha^t}) \in \mathbb{G}_p^{t+1}$. The secret key is the knowledge of α.

The PCS functions for KZG are:

Commit. *Commit* (PK, ϕ) takes the public key PK and a polynomial ϕ of degree at most $t+1$. It returns $\mathbf{g}^{\phi(\alpha)}$ as the commitment to ϕ.

CreateWitness. *CreateWitness*(PK, ϕ, i) takes the public key PK, a polynomial ϕ and a point i and returns $(i, \phi(i))$ and a witness $\omega_i = g^{\psi(\alpha)}$ where $\psi(x) = \frac{\phi(x)-\phi(i)}{(x-i)}$. Intuitively, *CreateWitness* reveals the point $(i, \phi(i))$. This can be verified without revealing the entire polynomial ϕ.

VerifyEval. *VerifyEval*$(PK, C, i, \phi_i, \omega_i)$ takes the public key PK, a commitment C to a polynomial ϕ and a point (i, ϕ_i) with a witness ω_i and outputs a boolean. Intuitively, it checks that the point (i, ϕ_i) is a point of ϕ using the witness and the commitment.

Open. *Open*(ϕ) takes a polynomial ϕ and reveals it.

Verify. *Verify*(PK, C, ϕ) takes the public key PK, a commitment C and a polynomial ϕ. It checks that the commitment corresponds to the polynomial.

Our first contribution is a formalization of the above functions in Isabelle and a formal proof that an interaction between an honest committer and verifier always yields a correct result.

III. Formally Verifying Security Properties

Our main contribution is a formal proof of the KZG security properties. In contrast to the reduction style arguments in the pen-and-paper proof [14], we rewrite the security properties proofs using the rigorous game-based approach after Shoup [19]. These games are then reduced to hardness assumptions (e.g. Discrete Log) via so called game-hops (see Appendix VI).

In the theorem prover Isabelle, we represent games and game-hops as probability mass functions and relations between them. The easiest way to do this formally is using the Giry monad [10], [18]. With the monadic structure, we can formulate games as probabilistic algorithms and have a one-to-one translation between them.

In the following, let $\mathcal{A}$ denote a probabilistic polynomial time (PPT) adversary. Each property holds if and only if the probability of winning the property-game is negligible for any PPT adversary $\mathcal{A}$. We show the security properties typical for PCSs:

Binding. The binding property intuitively asks if an adversary can find a commitment C and two polynomials that verify for the same commitment C. We distinguish between **polynomial binding** (verifying and opening the whole polynomial) and **evaluation binding** (verifying witnesses for point-wise reveals). The corresponding games can be found in the Appendix VII.

Hiding. We define the hiding game:

$$\left(\begin{array}{r l} \phi \leftarrow & \text{unif. random from } \{\phi \in \mathbb{Z}_p[x] \text{ s.t. } \deg(\phi) \leq t\}, \\ PK \leftarrow & \text{KeyGen}, \\ C = & \text{Commit}(PK, \phi), \\ wtnss = & \{\text{CreateWitness}(PK, \phi, i) \text{ for } i \in I\}, \\ \phi' \leftarrow & \mathcal{A}(PK, C, wtnss), \\ \text{return } \phi = \phi' & \end{array}\right)$$

The adversary wins if he can extract the polynomial ϕ from the commitment C and t witness tuples. Note that the evaluations of ϕ are at t arbitrary points as I is arbitrary.

IV. Proofs

The proofs for Polynomial Binding and Evaluation Binding in [14] can be adapted to a game-based proof quite easily. They reduce to the t-Strong Diffie-Hellmann assumption [14].

However, the hiding game and its proof cannot be deduced trivially. The proof for hiding given in [14] is a reduction style argument to the Discrete Log (DL) assumption. The main idea is to sample t random points $(i, \phi(i))$, turn them into group coordinates $(i, \mathbf{g}^{\phi(i)})$, add the DL-instance at the value 0, and interpolate $\mathbf{g}^{\phi}$ to extract the commitment $C = \mathbf{g}^{\phi(\alpha)}$. The main problem in the formalization of the pen-and-paper proof is that the original proof samples the indices $i \in I$ uniformly at random whereas the hiding game requires I to be an arbitrary list (not necessarily uniformly random). In conclusion, we cannot construct a game-based proof from this reduction trivially.

We propose two changes in the reduction: Firstly, we change the distribution of the set I from uniformly random to arbitrary to match the hiding game definition. For an arbitrary random list I, we still sample the evaluations $\phi(i)$ uniformly random. Secondly, instead of using the DL-instance at 0, we now choose a value x deterministically such that $x \notin I$ and use the DL-instance at x.

The second change is necessary for the correctness of the reduction. To interpolate a polynomial, the evaluation points must be distinct. The probability of 0 colliding with any of the t uniformly random chosen elements of I is negligible (due to the parameter choices). However, we cannot easily express the probability of 0 colliding with values in the arbitrary list I. We circumvent reasoning about probability and use a deterministic algorithm to choose a value x for the DL-instance, such that $x \notin I$.

V. Conclusion

Our formalization in Isabelle is (to our knowledge) the first formalization of a polynomial commitment scheme in a theorem prover. We formalized the specification of KZG and verified the correctness as well as the polynomial and evaluation binding. For the formalization of the security properties, we needed to rewrite the original reduction proofs as game-based proofs. During the formalization of the hiding property, we resolved issues with the translation to game-based proofs.

This ongoing effort of formalizing the KZG is a first step towards verifying blockchain security primitives.

Acknowledgment

We would like to thank Ian Goldberg for his kind explanations and help during the formalization process. Thanks also to Tobias Nipkow for his continuous support throughout this project.

References

[1] M. R. Albrecht, B. Dowling, and D. Jones. Device-oriented group messaging: A formal cryptographic analysis of matrix' core. Cryptology ePrint Archive, Paper 2023/1300, 2023. https://eprint.iacr.org/2023/1300.

[2] G. Arnon, A. Chiesa, G. Fenzi, and E. Yogev. Stir: Reed–solomon proximity testing with fewer queries. Cryptology ePrint Archive, Paper 2024/390, 2024. https://eprint.iacr.org/2024/390.

[3] D. A. Basin, A. Lochbihler, and S. R. Sefidgar. Crypthol: Game-based proofs in higher-order logic. Cryptology ePrint Archive, Paper 2017/753, 2017. https://eprint.iacr.org/2017/753.

[4] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev. Fast reed-solomon interactive oracle proofs of proximity. In I. Chatzigiannakis, C. Kaklamanis, D. Marx, and D. Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 14:1–14:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[5] E. Ben-Sasson, L. Goldberg, S. Kopparty, and S. Saraf. Deep-fri: Sampling outside the box improves soundness. Cryptology ePrint Archive, Paper 2019/336, 2019. https://eprint.iacr.org/2019/336.

[6] B. Bünz, B. Fisch, and A. Szepieniec. Transparent snarks from dark compilers. In A. Canteaut and Y. Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020*, pages 677–706, Cham, 2020. Springer International Publishing.

[7] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. Bulletproofs: Short proofs for confidential transactions and more. Cryptology ePrint Archive, Paper 2017/1066, 2017. https://eprint.iacr.org/2017/1066.

[8] D. Cadé. Cryptoverif. https://bblanche.gitlabpages.inria.fr/CryptoVerif/. last accessed: 08-04-2024.

[9] G. Couteau, T. Peters, and D. Pointcheval. Removing the strong rsa assumption from arguments over the integers. In J.-S. Coron and J. B. Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, pages 321–350, Cham, 2017. Springer International Publishing.

[10] M. Giry. A categorical approach to probability theory. In B. Banaschewski, editor, *Categorical Aspects of Topology and Analysis*, pages 68–85, Berlin, Heidelberg, 1982. Springer Berlin Heidelberg.

[11] GitHub. Cryptoline. https://github.com/fmlab-iis/cryptoline. last accessed: 08-04-2024.

[12] GitHub. EasyCrypt. https://github.com/EasyCrypt/easycrypt, 2022. last accessed: 06-05-2024.

[13] F. Huch. Archhive of formal proofs. https://www.isa-afp.org/, May 2024. last accessed: 02-05-2024, ISSN: 2150-914x.

[14] A. Kate, G. M. Zaverucha, and I. Goldberg. Constant-size commitments to polynomials and their applications. In *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security*, volume 6477 of *Lecture Notes in Computer Science*, pages 177–194. Springer, 2010.

[15] R. Maleckas, K. G. Paterson, and M. R. Albrecht. Practically-exploitable vulnerabilities in the jitsi video conferencing system. Cryptology ePrint Archive, Paper 2023/1118, 2023. https://eprint.iacr.org/2023/1118.

[16] T. Nipkow and G. Klein. *Concrete Semantics with Isabelle/HOL*. Springer, 2014. http://concrete-semantics.org.

[17] T. Nipkow, L. Paulson, and M. Wenzel. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer, 2002.

[18] nLab authors. Giry monad. https://ncatlab.org/nlab/show/Giry+monad, May 2024. Revision 72, last accessed: 02-05-2024.

[19] V. Shoup. Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Paper 2004/332, 2004. https://eprint.iacr.org/2004/332.

VI. Game-Hops in the KZG Security Proofs

In the KZG security proofs, we use three kind of game-hops:

1) Bridging Steps: Restating the game in an equivalent way.
2) Game-hops based on failure events: Two games are equivalent except for a specific failure occurring with negligible probability.
3) Over-estimations: Dropping a condition that needs to hold for the original game, thus obtaining a game with a higher winning probability for the adversary.

VII. Games for Polynomial and Evaluation Binding

Polynomial Binding. We define the polynomial binding game:

$$\left(\begin{array}{r} PK \leftarrow \text{KeyGen}, \\ (C, \phi, \psi) \leftarrow \mathcal{A}(PK), \\ b = \text{Verify}(PK, C, \phi), \\ b' = \text{Verify}(PK, C, \psi), \\ \text{return } \phi \neq \psi \wedge b \wedge b' \end{array}\right)$$

The adversary wins if he can find a commitment C and two polynomials that are a verifiable opening for C.

Evaluation Binding. We define the evaluation binding game:

$$\left(\begin{array}{r} PK \leftarrow \text{KeyGen}, \\ (C, i, \phi_i, \omega_i, \phi_i', \omega_i') \leftarrow \mathcal{A}(PK), \\ b = \text{VerifyEval}(PK, C, i, \phi_i, \omega_i), \\ b' = \text{VerifyEval}(PK, C, i, \phi_i', \omega_i'), \\ \text{return } \phi_i \neq \phi_i' \wedge b \wedge b' \end{array}\right)$$

The adversary wins if he can find a commitment C and a value i with two different claimed evaluation values, ϕ_i and ϕ_i', and according witnesses, ω_i and ω_i', such that the points (i, ϕ_i) and (i, ϕ_i') both verify.