**Grammar Study**

|  |  |  |
| --- | --- | --- |
| **Grammatical feature** | **Context** | **Translation** |
| **ARTICLES** | | |
| *“A/An”- numerical meaning of oneness* | 1. Frequently, users on the web need to show that they are, for example, not **a robot**, old enough to access an age restricted video, or eligible to  download an ebook from their local public library without being tracked. [1:1]  2. Here, both and could either be **a vector** of group elements or a vector of field elements. [3:8]  3. It also proposes **a solution** for higher-degree gates in the future work section but without security proof. [5:4] |  |
| *“Zero article” - with uncountable nouns* | 1. Therefore, parties do not need to publish any **information** besides its public key. [3:10]  2. The drive is assumed to be ‘truthful’ in that it simply relays **information** about x and y as queried. [4:12]  3. An instantiation of zk-creds requires a publicly accessible bulletin board to distribute the credential list, as well as parties running our **software**. [1:24] |  |
| *“A/An”- generic function* | 1. Scenario 2: **A cryptographer** walks into a bar. [1:27]  2. As a baseline, we can view the communication model discussed up  until this point as **a simple direct access model**. [4:12]  3. **A special case** of this relation, used in a number of zero knowledge protocols, is known as the polynomial zero check. [4:15] |  |
| *“Zero article” - generic function* | 1. We now build a system for in-person age verification coupled with photographic verification. [1:27]  2. Note that practical implementations of these protocols often involve careful considerations around **communication models** or cryptographic assumptions, which we elide here. [4:2]  3. There have been efforts to build accumulation **schemes** that overcome the limitations of fixed R1CS. [5:4] |  |
| *“The”- generic function* | 1. Moreover, in **the event** the user has trustworthy secure hardware, they can self-issue the credential by attesting the hardware ran this notary check itself. [1:30]  2. We compress **the** prover **message** by committing to them in a homomorphic commitment scheme. [5:4]  3. We know that **the rate** of decay is no worse than that for parallel repetition of the MIP projection, and sometimes it equals that. [2:7] |  |
| *“A/An”- first mention + “The” - second mention* | 1. Authority to issue **a credential** can be shown via zk-supporting-documentation that is itself the show of another credential. Moreover, because the proof in zk-supporting-documentation is general-purpose, the delegation process can constrain attributes in **the credential** being issued. [1:19]  2. Observe that, if **a user** has a valid Merkle authentication path (i.e., witness) attesting to their credential’s issuance at time , not all nodes in will usually need updating by time . Instead, **the user** only needs a summary of all Merkle tree nodes which have been added since time . [1:43]  3. The recursive circuit of this transformation is dominated by only scalar multiplications in the additive group of the commitment scheme for **a protocol** with prover messages and a degree d verifier. For R1CS, where and , this yields **the** same **protocol** and efficiency as Nova. [5:5] |  |
| *“A/An”- when modified by a descriptive attribute* | 1. Concretely, identity assertions using zk-creds take less than 150ms in **a real-world** scenario of using a passport to anonymously access age-restricted videos. [1:1]  2. Finally, given the accumulation scheme, if the relation R is NP-complete, we can apply the compiler in [BCLMS21] to obtain **an efficient** IVC scheme with predicates expressed in R. [5:9]  3. Instead, our work shows that one can directly use parallel repetition for PCPs to reduce the soundness error to **an arbitrary** constant while preserving the query complexity q. [2:7] |  |
| *“The”- when modified by a limiting attribute* | 1. In contrast to scanning the user’s driver’s license, this reveals only the minimal information necessary. [1:27]  2.  3. |  |
| *“The” - unique objects and notions* | 1. For this scenario, **the only issuer** is our passport-based issuer, and the access criteria being proved are age, expiry, and non-cloning. [1:26]  2. We consider an intermediate hybrid, where **the only difference** is in what is the aggregated signature and how the verification works. [3:36]  3. Fortunately, we observe that the syntactic mismatch of the messages sent by V and those sent by V′ is **the only issue** preventing us from concluding that MIP projection and PCP evaluation are inverses of each other. [2:31] |  |
| *“The” by reason of locality* | 1. This construction builds on **the results above**. [2:46]  2. We will focus almost universally on **the first part**: taking a large claim and reducing it to a much smaller one, such that it suffices only to verify the smaller claim, which, in turn, implies the original one with high probability. [4:13]  3. The relation statement can also add additional constraints on pc depending on **the applications**. [5:10] |  |
| *“The” when followed by an ordinal number* | 1. After **the first time**, the proof can be reused arbitrarily, until the user’s Merkle tree is updated by a new issuance. [1:29]  2. For example, the PCP for graph 3-coloring described above has soundness error less than 1 but its MIP projection has soundness error 1 (the first MIP prover always answers color 0 and **the second MIP prover** always answers color 1, regardless of the messages sent by the MIP verifier). [2:11]  3. **The first row** displays the native operations of the IVC prover. [5:27] |  |
| *“Zero article” when the noun is followed by a cardinal number* | 1. This computation requires FFT (as noted in **Lemma 1**) and, hence, the aggregation time is . [3:36]  2.  3. |  |
| **ACTIVE VOICE** | | |
| *Present Simple* | 1. **We provide a toolchain** to convert a passport into an anonymous credential. [1:26]  2. **Parallel repetition refers to a set** of valuable techniques used to reduce soundness error of probabilistic proofs while saving on certain efficiency measures. [2:1]  3. **The scheme does not require a** trusted **setup** or pairings, and the prover does not need to compute any FFTs. [5:1] |  |
| *Present Continuous* | 1. However, in the current application, since **we are delegating** computation of a zkSNARK, we require the server’s output to be a publicly verifiable zkSNARK. [3:5]  2. In reality, there are multiple trust sources for a given identity attribute, their credentials have distinctively different formats, and many, if not all, **issuers are unwilling to adopt** new protocols. [1:1]  3. |  |
| *Present Perfect* | 1. Thesepractical **constructions have opened a floodgate** of new applications. [3:3]  2. **We have used the fact** that in this bound. [4:11]  3. **We have discussed** conceptually simple access control **criteria** such as “my credential is not expired,” or “I am of age,” perhaps with a cryptographically complex mechanism for clone resistance. [1:28] |  |
| *Present Perfect Continuous* | 1.  2.  3. |  |
| *Past Simple* | 1. Prior to this work, **there were no known** black-box feasibility **results** for any of these applications. [3:1]  2. Note that, in the above construction, **we did not make use** of the freedom that many parameters, such as the distances between different elements, could be arbitrarily chosen. [4:33]  3. |  |
| *Past Continuous* | 1.  2.  3. |  |
| *Past Perfect* | 1.  2.  3. |  |
| *Future Simple* | 1. In this paper, **we will take** the linear-algebraic standard for notation (versus, e.g., some standards in coding theory). [4:2]  2. **A common view will be to look** at one specific element of the resulting vector Ax  3. |  |
| *Future Perfect* | 1.  2.  3. |  |
| **PASSIVE VOICE** | | |
| *Passive Voice (Present Simple)* | 1. Our polynomial commitment **is based on** the standard FRI-based polynomial commitment scheme. [3:4]  2.  3. |  |
| *Passive Voice (Present Continuous)* | 1. At a high level, this polynomial IOP performs the following operations on the extended witness (let R be the relation for which the proof **is being generated**). [3:28]  2.  3. |  |
| *Passive Voice (Present Perfect)* | 1. **These have been run** with hundreds of users and used to secure billions of dollars in cryptocurrency. [1:19]  2. Parallel **repetition has been studied** for interactive proofs (IPs) and multi-prover interactive proofs (MIPs). [2:1]  3. The problem of verifying privately delegated **computation has been studied** in prior works. [3:8] |  |
| *Passive Voice (Past Simple)* | 1. Michael Rosenberg’s work **was supported** by the National Defense and Engineering Graduate (NDSEG) Fellowship. [1:33]  2. Anonymous credentials **were developed** to address these concerns. [1:1]  3. Finally, we remark that before this work, the only black-box construction of a weighted threshold signature without setup was proposed by Micali et al. [3:7] |  |
| *Passive Voice (Past Perfect)* | 1.  2.  3. |  |
| *Passive Voice (Future Simple)* | 1. In particular, users can (eventually) rely solely on broadcasted frontier nodes to update their authentication path θ, the Merkle forest scales effectively as the number of credentials grows, and **every user will be required** to update their credentials’ proofs of membership less often compared to a naïve Merkle tree construction of comparable capacity. [1:46]  2.  3. |  |
| **VERBALS** | | |
| *Infinitive as a Subject or Attribute* | 1. We can use the rotation operation **to create** equivalence classes for verifier randomness, each  class of size n. [2:43]  2.  3. |  |
| *Infinitive as a Predicate* | 1. A common view will be **to look** at one specific element of the resulting vector Ax. [4:3]  2.  3. |  |
| *Infinitive as an Adverbial Modifier* | 1. However, **to ensure** the privacy of the secret witness employed in the generation of zkSNARK, **it is** highly **desirable** or, in many cases, necessary that this delegation remains privacy-preserving. [3:15]  2.  3. |  |
| *Complex Object with Infinitive* | 1. If n is large (say ) and , yet **we wish the probability** of failure **to be** no more than 2−100 then, repeating the experiment times suffices. [4:11]  2.  3. |  |
| *Complex Subject with Infinitive* | 1. Clearly, this requires the verifier **to run** in time linear in . [3:11]  2. Indeed, we did not optimize for any constants at all, and, using the bounds presented here, the proof system would be **unlikely to be useful** for practical applications. [4:33]  3. |  |
| *Gerund as a Subject* | 1. Given issued credentials via passports, **building** a privacy-preserving age verification scheme with zk-creds **is straightforward** and requires no new cryptography: website developers need simply define the issuers they will accept and construct the access criteria they need using gadget. [1:26]  2. **Computing** this proof **takes** less than 2 seconds. [1:24]  3. |  |
| *Gerund as an Adverbial Modifier* | 1. We associate to any given CSP a corresponding “canonical” PCP: the PCP string is the assignment to the variables of the CSP, and the PCP verifier samples a random constraint of the CSP and checks if it is satisfied (**by reading** from the PCP string the variables involved in that constraint). [2:5]  2.  3. |  |
| *Gerund as an Object (after preposition)* | 1. Finally, an issuer is able to revoke a credential if need be **by** simply **removing** it from the list. [1:14]  2. At a high level, we may view the above check as: we begin **by wishing to check** that some vector is q-close to a subspace (of whatever dimension) embedded in Fn. [4:32]  3. |  |
| *Gerund after Verbs (avoid, be worth, consider, finish, involve, allow, enable etc.)* | 1. Because zk-creds supports general purpose zero-knowledge proofs, geocoding restrictions are made more feasible with Groth16 gadgets: even if the Groth16 proof for the gadget is expensive, the resident or an outsourced prover **avoids recomputing** it every show. [1:29]  2.  3. |  |
| *Gerundial Complex* | 1. The benefit to this kind of customization is twofold: users can precompute and cache standalone gadget proofs separately from other access criteria proofs, and verifiers are freed **from having to define** custom circuits and generate the CRSs. [1:17]  2.  3. |  |
| *Participle as an Attribute* | 1. We will use some of the checks presented above to present a protocol that successively reduces checking that some (potentially very large) vector is close to a subspace, to checking that an **appropriately-chosen** smaller vector is close to a vector subspace. [4:27]  2. The parameters **presented** above are relatively abstract. [4:33]  3. One particular example we will use constantly is the **vector space consisting of the evaluations** of a polynomial of small degree. [4:4] |  |
| *Participle as an Adverbial Modifier* | 1. Every issuer has some issuance criteria ι that the requester must meet **in order to have their cred issued**, e.g., that the birth date in cred matches a signed digital passport. [1:12]  2.  3. |  |
| *Absolute Participial Construction* | 1. However, in our current application, since the computation **being delegated to the server** is the generation of a zkSNARK, the resulting zkSNARK must, therefore, be publicly verifiable. [3:16]  2.  3. |  |
| *Complex Object with the Participle* | 1.  2.  3. |  |
| *Complex Subject with the Participle* | 1.  2.  3. |  |
| **MODAL VERBS** | | |
| *Can / could / be able to* | 1. Note that, in the above construction, we did not make use of the freedom that many parameters, such as the distances between different elements, **could be arbitrarily chosen**. [4:33]  2. The tracking and data exposure risks raised by such requirements **can be eliminated** with privacy-preserving cryptography: anonymous credentials allow a user to assert that they meet some access criteria, e.g., are over 18, without revealing anything else about themselves, linking their viewing habits to their identity, or even linking distinct video views together. [1:2]  3. |  |
| *May / might* | 1. Therefore, parties **may sample** their key pairs independently, and no interactive/trusted setup is required. [3:16]  2.To understand succinct proofs, **we may contrast** them with ‘traditional’ computational proofs; i.e., providing a witness for a given statement. [4:1]  3. Traditional computational proofs **may be viewed** as a certificate that a certain computation was performed correctly. [4:1] |  |
| *Must / have to* | 1. Moreover, these proofs **must often be generated** obliviously, i.e., without knowledge of the secret. [3:1]  2.  3. |  |
| *Should / ought to* | 1. The proof size **should be sublinear** in the witness size. [3:18]  2.  3. |  |
| **CONDITIONALS** | | |
| *Zero Conditional* | 1. We remark that, **if we want** the above protocol to be publically verifiable, **then we** crucially **require** the underlying LHEncap scheme to be linearly-homomorphic w.r.t. randomness. [3:13]  2. As a special case, if the underlying LHEncap is deterministic, the above construction is always publicly verifiable. [3:13]  3. |  |
| *I Conditional* | 1.  2.  3. |  |
| *II Conditional* | 1.  2.  3. |  |
| *III Conditional* | 1.  2.  3. |  |

**References:**

1. Michael Rosenberg et al. (2022). zk-creds: Flexible Anonymous Credentials from zkSNARKs and Existing Identity Infrastructure. *Cryptology ePrint Archive*. URL: <https://eprint.iacr.org/2022/878.pdf>.
2. Alessandro Chiesa et al. (2023). On Parallel Repetition of PCPs. *Cryptology ePrint Archive*. URL: <https://eprint.iacr.org/2023/1714.pdf>.
3. Sanjam Garg et al. (2023). How to Prove Statements Obliviously? *Cryptology ePrint Archive*. URL: <https://eprint.iacr.org/2023/1609.pdf>.
4. Alex Evans. (2023). Succinct Proofs and Linear Algebra. *Cryptology ePrint Archive*. URL: <https://eprint.iacr.org/2023/1478.pdf>.
5. Benedikt Bunz. (2023). Protostar: Generic Efficient Accumulation/Folding for Special-sound Protocols. *Cryptology ePrint Archive*. URL: <https://eprint.iacr.org/2023/620.pdf>.