Anonymous Domain Ownership

Jan Lauinger, Jens Ernstberger, Sebastian Steinhorst
Technical University of Munich
Munich, Germany
name.surname@tum.de

***Abstract*—Emerging identity systems in the Web 3.0 achieve a new notion of digital ownership and control. Yet, existing identity ecosystems face bootstrapping issues as trustworthy authorities cannot be represented with adequate privacy.**

To equip authorities with trustworthy attributes, this work introduces a protocol of domain ownership oracles, where domain owners can prove domain ownership to any third party while remaining anonymous. To prove ownership of an anonymous domain *example.com*, users convince verifiers of the facts that (i) *example.com* is in control of users and that (ii) *example.com* belongs to a public anonymity set of domains. Verifiers learn nothing beyond these two facts and cannot determine which user owns which domain. For the first time, our work proposes a practical system to attribute anonymous domain owners with credentials of domain ownership where the degree of anonymity depends on the size of a public anonymity set of domains.

***Index Terms*—Domain Ownership, Blind Certificates, Oracles, Anonymity, Zero-knowledge Proofs.**

I. Introduction

Decentralized identity systems operate on the premise that users' credentials are attested to by issuers, allowing users' credentials to be validated without contacting the issuers [1]. Credential verifiers resolve auxiliary data from decentralized networks to validate claims that have been certified by issuers. Even though research around decentralized identity systems has enjoyed considerable attention throughout the past, actual deployments of decentralized credential systems remain sparse [2]. An explanation to the bootstrapping issue of decentralized credential systems is the fact that decentralized identity systems assume reputable issuers or authorities, which in reality, do not exist [3]. As a result, research has shifted to establish reputable identities, without a specific focus on representing trustworthy authorities.

Aiming to solve the bootstrapping issue of decentralized identity systems, we present a protocol to build trustworthy and reputable issuers. To bootstrap reputable issuers, we consider to attribute issuers with domain ownership credentials. Domain ownership is of core interest as (i) most authorities on the traditional Internet own domains or subdomains, and (ii) authorities are challenged to prove domain ownership to obtain X.509 certificates [4]. X.509 certificates are credentials of the public key infrastructure (PKI), which is the core trust infrastructure of today's Web. Further, we introduce anonymous domain ownership credentials, as not all organizations are eager to disclose their identity when attesting user claims.

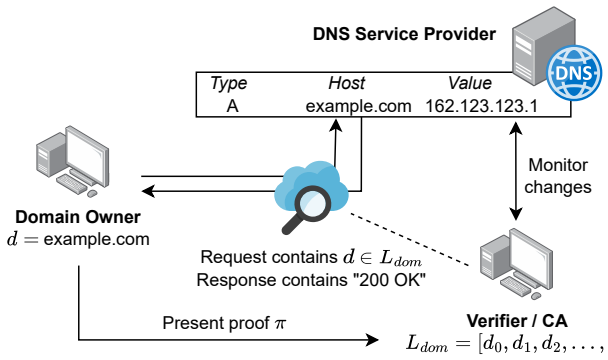

Fig. 1. High-level overview of the anonymous domain ownership protocol, where domain owners remain anonymous when proving domain ownership. Domain owners (i) update a DNS record of domain d with request r_{req}, (ii) receive a response r_{res} with the success status code, and (iii) send a zkSNARK proof π to the CA. The CA acts as a DECO proxy and captures TLS ciphertext traffic transcripts c_{tts} when tunneling DNS record updates of the domain owner. A CA issues domain ownership credentials if the verification of π asserts that $d \in L_{dom}$, $d \in r_{req}$, "200 OK" $\in r_{res}$, and $d \in \text{Dec}(k_{tls}, c_{tts})$, where k_{tls} is the TLS application traffic key negotiated in the TLS handshake between the DNS provider and the domain owner.

II. Protocol Overview

The main observation behind the anonymous domain ownership protocol is that the proxy mode setting described in the DECO protocol [5] can be applied to the domain validation challenge DNS-01, which is performed by X.509 certificate issuers in the PKI [4]. Inspired by the approaches found in [6], [7], we adapt the DNS-01 challenge such that the CA acts as a proxy between the domain owner and a DNS provider (cf. Figure 1). Similar to how the CA takes the role of issuing X.509 certificates in a PKI, the CA in the anonymous domain ownership protocol issues domain ownership credentials. In fact, we envision trusted PKI CAs as ideal candidates to adopt the anonymous domain ownership protocol, and with that, issue domain ownership credentials in the future.

Initially, the CA has access to a public list of domains L_{dom}, where the size of the list L_{dom} determines the degree of anonymity a domain owner can achieve. To obtain a credential of anonymous domain ownership, domain owners must first interact with an Internet Protocol (IP) anonymization network (e.g. Tor [8]), which randomizes source addresses and prevents address linking of clients (cf. relay network in Figure 2). Next, domain owners establish a TLS session with the DNS provider by tunneling through the CA. To convince a CA of owning domain d, domain owners are required to execute two main

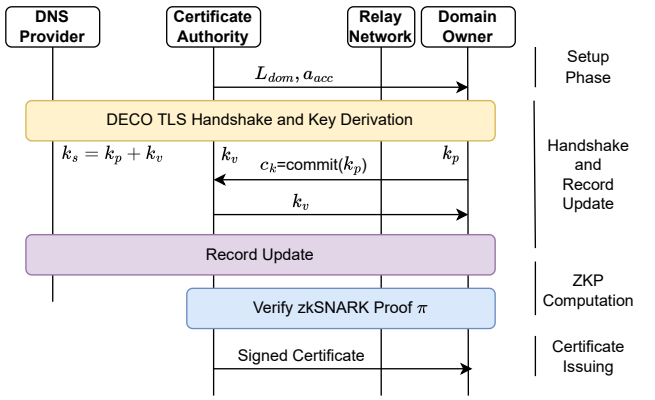

Fig. 2. Division of the anonymous domain ownership sequence diagram into the phases of setup, handshake, proof, and credential issuing. All connections from the domain owner are made through the anonymization relay network. First, the domain owner performs a DECO 3PHS with the CA and the DNS provider as the server. After the 3PHS, the domain owner and CA exchange their key shares, where the domain owner discloses a commitment of the key share first. Next, the domain owner updates a domain record at the DNS provider. The CA captures the record layer traffic transcript and verifies the private domain ownership proof from the domain owner. Upon successful verification of the zk-SNARK proof, the CA returns a blind credential.

steps. First, domain owners perform a domain record update at a DNS provider (cf. purple box in Figure 2). During a domain record update, the CA as the verifier acts as a proxy and forwards requests and responses between the domain owner and the DNS provider. Since the verifier intercepts traffic transcripts protected by TLS, it is up to the domain owner to prove that the protected TLS data complies with domain ownership requirements. As such, in a second step (cf. blue box in Figure 2), domain owners prove that the request of the domain record update contains the owned domain d and that the response contains a success status code (e.g. 200 OK). The success status code indicates a successful record update and ensures that the DNS provider accepts the requesting client as the domain owner. Further, the domain owner proves to the verifier that the owned domain d belongs to the public list of domains L_{dom} without revealing anything else. By using zero-knowledge proof technology to convince the verifier, verifiers learn nothing besides the fact that the domain d is a member of the set L_{dom} and verifiers cannot recall which domain $d \in L_{dom}$ domain owners own. Since domain record updates overwrite data with the same information, verifiers cannot observe changes among domain records and link or de-anonymize domain owners.

Further, to preserve *prover-integrity* [5], the domain owner must show that application traffic keys, which have been derived in the TLS handshake between the domain owner, CA, and DNS provider (cf. yellow box in Figure 2), correspond to symmetric encryption keys of the TLS record layer. *Prover-integrity* prevents malicious domain owners from equivocating domain ownership. To hide session keys while proving correct usage, domain owners must compute the record layer encryption of DNS record update requests and responses under the respective session keys in the zkSNARK circuit.

zk-circuit(k_p^{req}, k_p^{res}, d_{aux}^{req}, d_{aux}^{res}, r_d, w; $\mathbf{c}_k^{req}$, $\mathbf{c}_k^{res}$, $\mathbf{k}_v^{req}$, $\mathbf{k}_v^{res}$, $\mathbf{t}^{req}$, $\mathbf{t}^{res}$, $\hat{\mathbf{m}}_{req}$, $\hat{\mathbf{m}}_{res}$, **i**, **j**, $\mathbf{pt}_s$, $\mathbf{pt}_e$, $\mathbf{a}_{acc}$, $\mathbf{c}_d$):

1: $\mathbf{c}_k^{req} \stackrel{?}{=} \mathrm{Commit}(\mathrm{k}_p^{req})$; $\quad \mathbf{c}_k^{res} \stackrel{?}{=} \mathrm{Commit}(\mathrm{k}_p^{res})$
2: $\mathrm{k}^{req} \leftarrow \mathrm{k}_p^{req} \oplus \mathbf{k}_v^{req}$; $\quad \mathrm{k}^{res} \leftarrow \mathrm{k}_p^{res} \oplus \mathbf{k}_v^{res}$

3: $\mathbf{t}^{req} \stackrel{?}{=} \mathrm{AuthTag}(\mathrm{k}^{req}, \mathrm{d}_{aux}^{req})$; $\mathbf{t}^{res} \stackrel{?}{=} \mathrm{AuthTag}(\mathrm{k}^{res}, \mathrm{d}_{aux}^{res})$
4: $\hat{\mathbf{m}}_{res} \stackrel{?}{=} \mathrm{Enc}(m_{res}, \mathrm{k}^{res})$; $\quad \hat{\mathbf{m}}_{req} \stackrel{?}{=} \mathrm{Enc}(m_{req}, \mathrm{k}^{req})$

5: $1 \stackrel{?}{=} \mathrm{StrMatch}(m_{res}, \mathbf{i}, \text{“200 OK”})$
6: $\mathrm{s}_d \leftarrow \mathrm{StrExtractor}(m_{req}, \mathbf{j}, \mathbf{pt}_s, \mathbf{pt}_e)$

7: $1 \stackrel{?}{=} \mathrm{Membership}(\mathrm{s}_d, w, \mathbf{a}_{acc})$; $\quad \mathbf{c}_d \stackrel{?}{=} \mathrm{Commit}(\mathrm{s}_d, r_d)$

Fig. 3. Logic of the zk-SNARK circuit to prove anonymous domain ownership when using TLS 1.3. We highlight public input variables in **bold** text.

Last, to enable issuance of anonymous domain ownership credentials, the zkSNARK circuit must show that the domain d encrypts to a commitment. The commitment can be used in the domain ownership credential because it hides the domain as an identifier [7]. After the CA issues the blind credential of anonymous domain ownership, only the domain owner with knowledge of the commit randomness is able to proof certificate ownership towards any verifier. The entire logic of the zkSNARK circuit is illustrated in Figure 3.

III. Preliminary Security Analysis

System roles are ***domain owners***, which possess domain names that are (i) registered at a domain name registrar and (ii) are resolvable at name servers of DNS providers. ***CAs*** act as administrators of publicly resolvable domain lists L_{dom} and ***DNS providers*** expose an API which allows to set domain records at name servers. Under the assumptions of secure communication channels with fresh randomness per TLS session, up-to-date DNS records (CA can resolve and connect to the correct IP address of the DNS provider), and network traffic which cannot be blocked indefinitely, it can be shown that the protocol achieves ***domain-anonymity*** (domain owner unlinkability against every network participant except the DNS provider) and ***prover-integrity***, while remaining secure against the adversaries: $\mathcal{A}_1$ (semi-honest) trying to learn more than the validity of a proven statement. $\mathcal{A}_2$ (semi-honest) intending to de-anonymize the domain owner by eavesdropping, linking, modifying, or decrypting existing messages in the protocol. $\mathcal{A}_3$ (semi-malicious) owning a domain and intending to convince the CA from the untrue fact of owning a different domain.

IV. Conclusion

We introduce a protocol to obtain credentials of anonymous domain ownership. It can be shown that the protocol is *practical* with anonymous domain ownership verification times in the range of seconds. Additionally, the protocol is *legacy-compatibility*, such that DNS providers do not require server-side changes to remain compatible. The protocol serves as a building block to instantiate privacy-preserving and trustworthy attributes for decentralized identity issuers.

REFERENCES

[1] W. W. W. Consortium *et al.*, "Verifiable credentials data model 1.0: Expressing verifiable information on the web," *https://www. w3. org/TR/vc-data-model/?# core-data-model*, 2019.

[2] D. Maram, H. Malvai, F. Zhang, N. Jean-Louis, A. Frolov, T. Kell, T. Lobban, C. Moy, A. Juels, and A. Miller, "Candid: Can-do decentralized identity with legacy compatibility, sybil-resistance, and accountability," in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021, pp. 1348–1366.

[3] M. Rosenberg, J. White, C. Garman, and I. Miers, "zk-creds: Flexible anonymous credentials from zksnarks and existing identity infrastructure," *Cryptology ePrint Archive*, 2022.

[4] J. Aas, R. Barnes, B. Case, Z. Durumeric, P. Eckersley, A. Flores-López, J. A. Halderman, J. Hoffman-Andrews, J. Kasten, E. Rescorla *et al.*, "Let's encrypt: an automated certificate authority to encrypt the entire web," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 2473–2487.

[5] F. Zhang, D. Maram, H. Malvai, S. Goldfeder, and A. Juels, "Deco: Liberating web data using decentralized oracles for tls," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 1919–1938.

[6] P. Grubbs, A. Arun, Y. Zhang, J. Bonneau, and M. Walfish, "Zero-knowledge middleboxes," *Cryptology ePrint Archive*, 2021.

[7] L. Wang, G. Asharov, R. Pass, T. Ristenpart, and A. Shelat, "Blind certificate authorities," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 1015–1032.

[8] K. Loesing, S. J. Murdoch, and R. Dingledine, "A case study on measuring statistical data in the Tor anonymity network," in *Proceedings of the Workshop on Ethics in Computer Security Research (WECSR 2010)*, ser. LNCS. Springer, January 2010.