Securing DNSSEC Keys via Threshold ECDSA From Generic MPC

Anders Dalskov[1], Marcel Keller[2], Claudio Orlandi[1],
Kris Shrishak[3], Haya Shulman[3,4]
[1]Aarhus University [2]CSIRO's Data61 [3]TU Darmstadt [4]Fraunhofer SIT

A surge in DNS cache poisoning attacks in the recent years generated an incentive to push forward the deployment of DNSSEC (Arends, Austein, Larson, Massey & Rose (2005)). ICANN accredited registrars are required to support DNSSEC signing for their customers, and the number of signed domains is slowly increasing. Yet with the increase in number of signed domains, the number of vulnerable DNSSEC deployments is also increasing. However, due to lack of support for other, more efficient algorithms, the most popular cryptographic algorithm is RSA. Furthermore, to avoid overhead, network operators typically use short keys (1024 bits) which are no longer considered secure (Chung, van Rijswijk-Deij, Chandrasekaran, Choffnes, Levin, Maggs, Mislove & Wilson (2017)).

In practice, few domain owners run and manage their own authoritative name servers, and this role is typically outsourced to the DNS operators. However, while this has its benefits (e.g., the name servers are then managed by experts), it introduces several issues related to key management. Domain owners have to relinquish control of their private signing keys, which makes the operators very attractive targets for attackers. Moreover, the operators can themselves include any information in the DNS responses as they have full control over their customer's domains. Finally, and in practice, operators typically reuse keys for multiple domains which further exacerbates the above issues.

In this work, we explore the use of ECDSA threshold protocols for DNSSEC zone signing. To that end, we present a *generic* transformation of any secure multiparty computation (MPC) protocol over $\mathbb{Z}_p$ to one that can perform threshold signing. We show that this transformation leads to *very* efficient signing protocols in the preprocessing model, which is attractive in the context of zone signing (because the MPC participants, i.e., the authoritative name server operators, can estimate how many signatures they need to create based on the amount of traffic a specific domain gets). Compared to existing specialized threshold signing protocols, our generic approach has the benefit that it is faster, and more importantly for real-world use, allows the operators to "tailor" protocols to a specific threat model (passive vs. active security and honest vs. dishonest majority). To further illustrate the usefulness of our work, we integrate our signing implementations which are implemented in the MP-SPDZ framework (Data61 (2019)) with an existing and widely used open-source DNS server application, Knot DNS (Knot (2019)). Finally, we remark that key generation is trivial using our technique as it amounts to generating a single shared value.

References

Roy Arends, Rob Austein, Matt Larson, Dan Massey & Scott Rose (2005). DNS Security Introduction and Requirements. *RFC* **4033**, 1–21.

Taejoong Chung, Roland van Rijswijk-Deij, Balakrishnan Chandrasekaran, David R. Choffnes, Dave Levin, Bruce M. Maggs, Alan Mislove & Christo Wilson (2017). A Longitudinal, End-to-End View of the DNSSEC Ecosystem. In *USENIX Security Symposium*, 1307–1322. USENIX Association.

Data61 (2019). MP-SPDZ - Versatile framework for multi-party computation. `https://github.com/data61/MP-SPDZ`.

Knot (2019). Knot DNS. `https://www.knot-dns.cz/`.

crypto day matters 31 (2019)
Beiträge vom 31. Kryptotag, 17.-18.10.2019, Qualcomm, Berlin.
doi:10.18420/cdm-2019-31-07