



not under the control of a malicious adversary. We take the position that it is necessary to develop techniques where it is possible to provide *trustworthy information services on untrusted machines*. In this light, we have developed techniques by which relational querying services can be provided by an untrusted publisher on behalf of a trusted data creator, or owner.

Our techniques allow an **untrusted publisher** to produce a **verification-object**  $\mathcal{VO}$  for a *user* along with the answer to her database query. The user can use the  $\mathcal{VO}$  to verify that the answer is identical to the one that *owner* of the database would provide. The verification-object is based on a **summary-signature** that the owner periodically distributes to the publisher. See Figure 1.

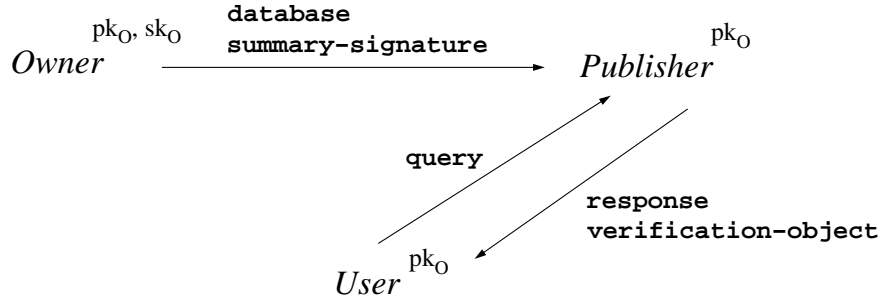


Figure 1: We partition the role of information provider into *owner* and *publisher*. The owner provides database updates and summary signatures to the publisher. The publisher is untrusted. The *user* makes inquiries of the publisher. She gets responses which can be verified using a returned *verification-object*. Superscripts denote keys known to that party. Only  $sk_O$  is secret. The user must be sure of the binding of  $pk_O$  to the owner (figure is from [2]). Another approach is for the *owner* to send the summary signature directly to *user*.

The summary-signature is based on a Merkle-tree like construction [4] over B-tree type indexes for the relations in the owner’s database; a hash digest of the entire tree is signed with  $sk_O$ . Verification objects are constructed by using the same B-tree. A  $\mathcal{VO}$  validates an answer by providing an unforgeable chain (or rather, sub-tree) of hashes, constituting a “proof”, which links the answer to the summary-signature. Our approach has several features:

1. In addition to his own security, a user need only trust the signing key of the owner. The owner only sends out new summary-signatures after database updates. This allows the owner’s private key to be kept “offline”, isolated from network-based attacks.
2. Users need not trust the publishers, nor their keys. Even if an attacker replaces a publisher’s copy of  $pk_O$  by a fictitious key, this would only result in a loss of service from him.
3. Our techniques provide for verification-objects whose size is linear in the size of the answer, and logarithmic in the size of the database, for a useful subset of standard relational queries.
4. The verification-object guarantees not only that all the information delivered in the answer was actually in the database, but also that no information was incorrectly left out.
5. In all of our techniques there is reasonable overhead for for computing the summary-signature, for computing the verification-object, and for verifying the verification-object.
6. The approach gives rise to a highly *survivable* system, since publishers can be replicated without co-ordination, and the loss of a publisher entails no degradation of system security.
7. The ability to use untrusted publishers enables *scalable* and *flexible* delivery of information.

A correct answer and verification-object will always be accepted by the user. An incorrect answer and verification-object will almost always be rejected, since our techniques make it computationally infeasible to forge a correct verification-object for an incorrect answer. Overall, the approach nicely simplifies the operational security requirements for both owners and publishers.

Full details, including related work, more specifics on the algorithms, and security theorems, can be found in a recent paper [2]. Currently, our approach works with databases using the relational model. We support selections, projections, unions, and a limited notion of joins. We also support a limited notion of intersections corresponding to multi-dimensional range queries. Work is underway to extend this work to cover more aspects of relational queries, as well as to other data models.

We also note here that the notion of authentic query-processing distinguishes our approach from peer-to-peer models such as Gnutella [6] and approaches for file replication [1, 8] which focus primarily on the monolithic publication of entire files, rather than on processing queries over their content.

## 2 Contribution to workshop

We propose a novel approach to information survivability, which allows the of *untrusted* publishers to offer replicated query-answering services. The use of these servers can be used to freely provide additional redundancy, without making the replicated service more vulnerable to attack.

We would be interested in participating in discussions on database survivability. We are also interested in general approach to performing trusted computations on untrusted machines. We have prior published work in this area [3], where we explored the storage of certain types of data-structures (stacks and queues) on an untrusted machine. We are also interested in the connection between authentic publication, and the problem of computing with secret data (E.g., [7])

## References

- [1] R. J. Anderson. The Eternity Service. *In Proceedings of Pragocrypt*, 1996.
- [2] Devanbu, P., Gertz, M., Martel, C., and Stubblebine, S., Authentic Third-Party Data Publication, *Fourteenth IFIP Working Conference on Database Security, (DBSEC 2000)*, Schoorl, The Netherlands, August 2000.
- [3] Devanbu, P., and Stubblebine, S., Stacks and Queues on untrusted platforms, *IEEE Symposium on Security and Privacy*, Oakland, USA, May 1998.
- [4] R.C. Merkle. A certified digital signature. In *Advances in Cryptology–Crypto '89*, 1989.
- [5] M. Naor, K. Nissim. Certificate Revocation and Certificate Update. *Proceedings, 7th USENIX Security Symposium*, 1998.
- [6] The Gnutella System <http://gnutella.wego.com>
- [7] Song, D., Wagner D., and Perrig, A., Search on Encrypted Data, *Proc. of Security and Privacy Symposium*, Oakland, USA. May 2000.
- [8] Waldman, M., Rubin, A., and Cranor, L. F., Publius: A robust, tamper-evident, censorship-resistant web publishing system <http://www.cs.nyu.edu/~waldman/publius/publius.pdf>