Privacy-Preserving Accountable Computation

Michael Backes
Saarland University and MPI-SWS
Saarbrücken, Germany
backes@cs.uni-saarland.de

Dario Fiore
MPI-SWS
Saarbrücken, Germany
fiore@mpi-sws.org

Esfandiar Mohammadi
Saarland University
Saarbrücken, Germany
mohammadi@cs.uni-saarland.de

In distributed systems, checking whether a node's operation is correct or faulty is a major concern. Indeed, faulty actions can occur for many reasons: a node can be affected by a hardware or software failure, a node can be compromised by an attacker, or a node's operator can deliberately tamper with its software. Detecting such faulty nodes is often very difficult, in particular for large-scale systems.

Recent work proposed *accountability* as a paradigm to ensure that whenever an incorrect behavior is observed, it can be linked to a malicious node. At the same time, honest nodes gain the ability to disprove any false accusations. Examples of these accountability systems include PeerReview [5] and its extension [1]. The basic idea of PeerReview is that every user generates a tamper-evident log which contains a complete trace of the performed computations. Later, an auditor (in PeerReview any other node in the distributed system) can check the correctness of the user's operations by inspecting the logs, replaying the execution of the user using a reference implementation, and finally comparing its result. The above approach is however restricted to deterministic systems. Indeed, in order to enable the replay of a randomized computation one should publish the seed of the pseudo-random generator in the logs. Clearly, this would completely destroy the unpredictability of future pseudo-random values. This issue was addressed by CSAR, an extension of PeerReview [1]. More specifically, the main contribution in [1] is to formalize a notion of accountable randomness, called *strong accountable randomness*, and to present the construction of a pseudo-random generator satisfying this property. Informally, strong accountable randomness consists of the following requirements: (i) the pseudo-random generator generates values that look random, even to the party who computes them; (ii) it is possible to verify that the values were computed correctly; (iii) the unpredictability of future values (i.e., those for which a proof was not yet issued) does not get compromised; and (iv) the above properties are fulfilled even if a malicious party is involved in the seed generation.

A full version of this paper can be found under the following address: http://www.infsec.cs.uni-saarland.de/~mohammadi/ppac.html.

While the approach of PeerReview and CSAR is very general and has been proven practical, these techniques have an inherent drawback: they inevitably expose a party's private data. In many scenarios such a privacy leak is unacceptable and might thus discourage the adoption of accountability systems. For instance, consider a company that runs its business using a specific software. There are many cases in which companies' tasks have to comply with legal regulations, and having a system which allows an auditor to check this compliance in a reliable way would be highly desirable. On the other hand, companies have a lot of data that they want to keep secret. This data might include, for instance, business secrets such as internal financial information, or secret keys for digital signatures or encryption schemes.

In spite of its utter importance, the idea of providing accountability while preserving the privacy of the party's data has not been yet properly explored in previous work.

A. Our contribution

We address this important open problem in the area of accountability providing three main contributions:

- We formalize a notion of privacy-preserving accountability for randomized systems. At a high level, our notion requires that a user is able to produce a log that convinces an auditor of the correctness of (1) the outcome of a computation (e.g., that $y = P(x)$), and (2) the generated randomness. At the same time, the contents of the log neither compromise the secrecy of specific inputs of the computation nor the unpredictability of the randomness generated in the future. Our notion is defined in the UC framework, and thus allows for arbitrary composability.
- We focus on efficient realizations of privacy-preserving accountability for randomized systems. To do so we identify appropriate cryptographic constructions. We use the non-interactive proof system by Groth and Sahai [3] which supports statements in the language of pairing-product equations, and a pseudorandom function, due to Dodis and Yampolskiy [2], which works

We are aware that the UC framework has flaws. Our results, however, can be straight-forwardly migrated to other simulation-based composability frameworks [8], [7], [6].

in bilinear groups and is thus compatible with this language. With the above proof system we can characterize a variety of computations: efficient solutions exist for the case of algebraic computations with equations of degree up to 2, but also arbitrary circuits can be supported [4].

- We present the first efficient accountable signature scheme: a scheme in which the signer can show that the secret key and the signatures are generated "correctly", i.e., by using accountable randomness. Without this property a malicious signer could indeed use malicious randomness, and then argue that this happened by bad luck in order to repudiate some signatures.

B. Our contribution in detail

In this section we give a high level explanation of the technical ideas and the approaches used in this paper.

Our notion and its relation with strong randomness: In the case of deterministic computations the notion of privacy-preserving accountability would essentially fall into the well-known application area of zero-knowledge proofs. However, we model randomized computations: consequently we want that even the randomness is accountable, i.e., correctly generated. While such a notion, called strong accountable randomness, has already been introduced in [1], we show that it is *not* realizable without random oracles.

Recall that strong accountable randomness requires that the pseudo-randomness of the generated values must hold even against the party who knows the seed. Clearly, this is a very strong property. A random oracle helps its realization as it essentially destroys any algebraic properties or relations that one may recognize in such values. But without the help of this "magic" tool, it is clear that the party computing the values knows *at least* how they were computed.

Our impossibility result left us with two opportunities: (1) either define privacy-preserving accountability for randomized computations in the strongest possible way (i.e., so as to imply strong randomness) but be aware that it would be realizable only using random oracles, or (2) define a slightly weaker version of accountable randomness. Although the first option would be preferable, a careful analysis revealed that its efficient realization is very unlikely. Indeed, any meaningful notion of privacy-preserving accountable computation fulfilling the properties we have in mind will need zero-knowledge proofs in order to be realized. At the same time, these proofs would have to involve a pseudo-random generator that satisfies strong randomness by using a hash function modeled as a random oracle. We are not aware of any hash function that allows for efficient zero-knowledge proofs and whose actual implementation maintains unpredictability properties close to the ideal ones of a random oracle (i.e., its use in a scheme does not fall prey to trivial attacks). This is why we decided to follow the second approach.

On realizing accountable signatures: While focusing on more specific applications of our accountability system, we asked how to efficiently prove statements that involve the randomness generated by our system. For instance, many cryptographic protocols rely on correctly distributed randomness, but such randomness usually cannot be revealed (thus CSAR is not a solution). In particular, this property is very interesting for digital signatures as it would allow for the accountability of this primitive, namely the signer could show that the secret key and the signatures are generated correctly, i.e., by using accountable randomness.

Towards this goal, the technical challenge is that for the combination of Groth-Sahai proofs and our specific pseudo-random generator random values that need to be hidden can only be group elements. We are not aware of any signature scheme, from the literature, in which *all* random values (e.g., the secret key and the randomness) can be computed using our pseudo-random generator. In this work we propose the construction of such a signature scheme which thus satisfies our notion of accountability.

References

[1] M. Backes, P. Druschel, A. Haeberlen, and D. Unruh, "Csar: A practical and provable technique to make randomized systems accountable," in *Proceedings of the Network and Distributed System Security Symposium (NDSS 2009)*. The Internet Society, 2009.

[2] Y. Dodis and A. Yampolskiy, "A verifiable random function with short proofs and keys," in *Public Key Cryptography - PKC 2005*, ser. Lecture Notes in Computer Science, S. Vaudenay, Ed., vol. 3386. Springer Berlin Heidelberg, 2005, pp. 416–431.

[3] J. Groth and A. Sahai, "Efficient noninteractive proof systems for bilinear groups," *SIAM Journal on Computing*, vol. 41, no. 5, pp. 1193–1232, 2012.

[4] J. Groth, "Short pairing-based non-interactive zero-knowledge arguments," in *Advances in Cryptology - ASIACRYPT 2010*, ser. Lecture Notes in Computer Science, M. Abe, Ed., vol. 6477. Springer Berlin Heidelberg, 2010, pp. 321–340.

[5] A. Haeberlen, P. Kuznetsov, and P. Druschel, "PeerReview: Practical accountability for distributed systems," in *Proceedings of the 21st ACM Symposium on Operating Systems Principles (SOSP'07)*, Oct 2007.

[6] D. Hofheinz and V. Shoup, "Gnuc: A new universal composability framework," *IACR Cryptology ePrint Archive*, p. 303, 2011.

[7] R. Küsters, "Simulation-based security with inexhaustible interactive turing machines," in *Proc19th IEEE Computer Security Foundations Workshop*. IEEE Computer Society, 2006, pp. 309–320.

[8] B. Pfitzmann and M. Waidner, "A model for asynchronous reactive systems and its application to secure message transmission," *Security and Privacy, IEEE Symposium on*, vol. 0, p. 0184, 2001.

In particular, every known pseudo-random function compatible with Groth-Sahai outputs group elements.