Efficient Accountable Multisignatures

Andrew Poelstra
apoelstra@blockstream.com *

2015-02-11 (commit 607815e)

Abstract

It is well-known that n-of-n Schnorr multisignatures can be produced in one round of communication by adding ordinary Schnorr signatures. As observed by Boneh, this can be extended from n-of-n to arbitrary monotone functions of the signers by use of a linear secret sharing scheme.

However, such signatures have the property that they are *signer indistinguishable*; that is, any signer set which is able to produce a signature produces one which is indistinguishable from that produced by any other. (In fact, without extra knowledge of the verification key structure, these signatures are indistinguishable from ordinary single-signer Schnorr signatures.) In some contexts, it is important for auditability to be able to determine which signer set produced a given signature.

We therefore study *accountable multisignatures*. The most straightforward way to do this is to define as a verification key the concatenation of all signers' verification keys along with a description of the admissible signer sets, giving $O(n)$ verification key size in the number of signers. We significantly improve this in many cases.

License. This work is released into the public domain.

*This work was partially supported by Blockstream.

1 Introduction

It is well-known that n-of-n Schnorr multisignatures can be produced in one round of communication by adding ordinary Schnorr signatures. Specifically, Schnorr signatures[Sch89] consist of a pair (s, R) where $s = k - xe$ and $R = kG$ where G is a generator of the underlying group, k is a nonce and x is a secret key. If n signers first publish k_iG to each other, they are each able to produce a signature (s_i, R) where $s_i = k_i - x_ie$ and $R = k_iG$. The the pair $(s, R) = (\sum s_i, R)$ is then a valid Schnorr signature of the message m with verification key $P = \sum xG$.

As observed by Boneh[1], this can be extended from n-of-n to arbitrary monotone functions of the signers by use of a linear secret sharing scheme. Specifically, each signer distributes shares of her x_i and k_i to every other signer. Then in the case that she does not participate in producing a signature, an admissible set of signers is able to construct $x_i - k_ie$ in her stead by applying the secret sharing scheme. (Note that the signers combine the shares of x_i and k_i to produces shares of $x_i - k_ie$ once they know e; thus neither x_i nor k_i is ever learned by anyone.)

However, the resultant signature is one with verification key $\sum_i x_iG$ and public nonce $\sum_i k_iG$, regardless of which signer set was used to produce it. This means that in contexts where knownledge of the signer set is needed after the fact (*e.g.* in an escrowed Bitcoin transaction where it may be of legal consequence whether the escrow agent was involved in moving some coins), these multisignatures are inappropriate.

In order to produce a signature for which the signer set can be identified, the most natural thing to do is to have each signer produce a verification key P_i, and publish the multisignature verification key as $\{P_i\}$ along with a description the admissible signer sets. Then a multisignature of a message m by a signer set S consists of individual signatures σ_i by each signer $i \in S$ along with a description of S. However, our verification key size is then linear in the total number of signers and the signature size is linear in the size of S. By putting the full keyset $\{P_i\}$ in a Merkle tree, the keysize can be improved to logarithmic in the number of signers n, at the cost of making signatures have size $n \log |S|$ (since each signer must provide a proof that her key is in the list).

An improvement to this scheme is to produce a n-of-n verification key $\sum_{i \in S} x_iG$ for every admissible set S, and publish these keys. For a simple k-of-n threshold multisignature there are $\binom{n}{k}$ admissible subsets, so by putting these keys in a Merkle tree we obtain a constant verification key size (just a Merkle root) and $\log \binom{n}{k}$ signature size.

However, this scheme requires the precomputation of $\binom{n}{k}$ sums of verification keys, which grows as a degree-k polynomial in n, which is prohibitive for cases as small as $n = 30$, $k = 15$.

Instead, we propose a scheme for threshold signatures in which the verification key consists of only $n - k + 1$ keys and signatures require only a list of $(n - k + 1)$ small integers to identify the signer set and its key. The way it works is essentially to publish a basis of the linear space spanned by the keys corresponding to every signer set, and for signatures to then identify keys by giving coefficients of linear combinations of this basis.

[1] Dan Boneh, personal communication, 2013.

2 Construction

As a first step we give the construction only for threshold signatures. Let $S = [1,n]$ be a set of n signers with individual keypairs $(x_i, P_i = x_i G)$. Suppose that any k of n signers are allowed to produce a signature. Then we compute $(n-k+1)$ points Q_i for $i = 0, \ldots, n-k$ as follows:

$$Q_i = \sum_{j=1}^{n} j^i P_i$$

Then let $S' \subseteq S$ be an admissible set of signers, *i.e.* $|S'| \geq k$. To produce a signature, they act as follows:

1. As $|S \setminus S'| \leq n-k$, they can compute a polynomial

   $$p(x) = c_{n-k} x^{n-k} + c_{n-k-1} x^{n-k-1} + \cdots + c_0$$

   such that $p(i) = 0$ exactly when $i \in S \setminus S'$. The signers compute

   $$Q = \sum_{i=0}^{n-k} c_i Q_i = \sum_{j=1}^{n} p(j) P_i$$

   We observe that this is a linear sum of the P_i's which has nonzero coefficient of P_i exactly when $i \in S$.

2. Each signer $i \in S'$ computes a random nonce k_i and sents $k_i G$ to the other signers.

3. They all compute $R = \sum_{i \in S'} p(i) k_i G$ and $e = H(m, e)$. Each one computes $s_i = x_i - k_i e$.

4. Then $(s, R) = (\sum_{i \in S'} p(i) s_i, R)$ is a valid signature with verification key Q.

The complete signature consists of the pair (s, R) along with a description of p.

2.1 Correctness

2.2 Security

References

[Sch89] C. P. Schnorr, *Efficient identification and signatures for smart cards*, Proceedings of CRYPTO '89, 1989, ftp://utopia.hacktic.nl/pub/mirrors/Advances%20in%20Cryptology/HTML/PDF/C89/239.PDF.