The MuSig Schnorr Signature Scheme

November 18, 2018

Contents

1 Introduction

This report investigates MuSig, which is provably secure in the *plain public-key model*. However, the case of interactive signature aggregation where each signer signs their own message must still be proven by a complete security analysis.

Multi-signatures are a form of technology used to add additional security for cryptocurrency transactions. A multi-signature protocol which allows a group of signers to produce a short, joint signature on a common message. [1]

2 Schnorr Signatures for Bitcoin

Schnorr signatures produce a smaller on-chain size, support faster validation and have better privacy. They natively allow for combining multiple signatures into one through aggregation. They permit more complex spending policies, including k-of-n and more to be represented as as single signature for a single key.

Signature aggregation also has its challenges. This included the rogue-key attack, where a participant steals funds using a specifically constructed key. This is easily solved for simple multi-signatures, however, through an enrollment procedure, where the keys sign themselves, supporting it across multiple inputs of a transaction requires plain public-key security, meaning there is no setup.

An additional attack, termed the Russel attacks, after Russel O'Connor, who was discovered for multi-party schemes where a party could claim ownership of someone else's key and so spend their other outputs.

Peter Wuille discussed the issues and their solutions, which refines the Bellare-Neven (BN) scheme. He also discussed the performance improvements that were implemented for the scaler multiplication fo the BN scheme and how they enable batch validation on the blockchain. A pair of BIPs are in process to make these advances a reality for Bitcoin.[2]

3 Key Aggregation for Schnorr Signatures

MuSig is a simple multi-signature scheme that is novel in combining:

1. Support for key aggregation;
2. Security in the plain public-key model.

There are two versions of MuSig, that are provably secure, which differ based on the number of communication rounds:

- Three-round MuSig only relies on the Discrete Logarithm (DL) assumption, on which ECDSA (Elliptic Curve Digital Signature Algorithm) also relies
- Two-round MuSig instead relies on the slightly stronger One-More Discrete Logarithm (OMDL) assumption

A multi-signature scheme is a combination of a signing and verification algorithm, where multiple signers (each with their own private/public key) jointly sign a single message, resulting in a single signature. This can then be verified by anyone knowing the message and the public keys of the signers.

Note: in the context of Bitcoin, the term 'multisig' refers to a k-of-n policy, where k can be different from n. While in the cryptographic literature, the term multi signature really only refers to n-of-n policies, however, k-of-n can be constructed on top of n-of-n.

The term *key aggregation* refers to multi-signatures that look like a single-key signature, but with respect to an aggregated public key that is a function of only the participants' public keys. Thus, verifiers do not require the knowledge of the original participants' public keys- they can just be given the aggregated key. In some use cases, this leads to better privacy and performance. MuSig is effectively a key aggregation scheme for Schnorr signatures.

There are other multi-signature schemes that already exist that provide key aggregation for Schnorr signatures, however they come with some limitations,

such as needing to verify that participants actually have the private key corresponding to the pubic keys that they claim to have. *Security in the plain public-key model* means that no limitations exist. All that is needed from the participants is their public keys. [3]

3.1 Applications of mulit-signatures in Bitcoin

The most obvious use case for mulit-signatures with regards to Bitcoin is as a more efficient replacement of n-of-n multisig scripts and other policies that permit a number of possible combinations of keys (including k-of-n, using key trees, MAST, or traditional threshold schemes). For these, a native multi-signature scheme means that what is left is one signature per transaction input.

A key aggregation scheme also lets us reduce the number of public keys per input to one, as a user can send coins to the aggregate of all involved key, rather than including them all in the script. This leads to smaller on-chain footprint, faster validation, and better privacy. As a result, MuSig is a good choice here.

Instead of creating restrictions with one signature per input, one signature can be used for the entire transaction. Key aggregation cannot be used across multiple inputs, as the public keys are committed to by the outputs, and those can be spent independently. MuSig can be used here (with key aggregation done by the verifier).

On a technical standing, in order to combine all the transaction inputs' signatures, a multi-signature scheme is not necessary, instead an aggregate signature scheme can be used. The distinction is simply that in an aggregate signature, each signer has their own message, instead of one message shared by all.

Aggregate signatures can be categorized as being:

- Interactive: Interactive aggregate signatures (IAS) require the signers to cooperate, while non-interactive schemes all the aggregation to be done by anyone
- Non-interactive: These allow the aggregation to be done by anyone

No non-interactive aggregation schemes are known that only rely on the DL assumption, but interactive ones are trivial to construct: where a multi-signature scheme has every participant sign the concatenation of all messages. The paper by Blocksteam, focusing on key aggregation for Schnorr Signatures shows that this is not always a desirable construction, and gives an IAS variant of BN with better properties instead. [3]

3.1.1 Details

Notation

- $x, x_1, x_{2...}$ are private keys with corresponding public keys $X, X_1, X_2 ... X_i = x_i G$, with G the generator
- The message being signed is m

- $H()$ is a cryptographic hash function

Schnorr Signatures

- Signatures are $(R, s) = (rG, r + H(X, R, m)x)$ where $r$is a random nonce chosen by the signer
- Verification requires $sG = R + H(X, R, m)X$

Naive Schnorr multi-signatures

- Call X the sum of the X_i points
- Each signer chooses a random nonce r_i and shares $R_i = r_iG$ with the other signers
- Call R the sum of the R_i points
- Each signer computes $s_i = r_i + H(X, R, m)x_i$
- The final signature is (R, s) where s is the sum of the $s_i$values
- Verification requires $sG = R + H(X, R, m)X$, where X is the sum of the individual public keys

It is interesting to note that this satisfies the definition of a *key aggregation scheme*, as multiple parties can jointly produce a signature that is a valid single-key signature for the sum of the keys.

The issue arises in that this scheme is not secure. Consider the following scenario:

- Alice and Bob want to produce a multi-signature together.
- Alice has a key pair (x_A, X_A) and Bob has $(x_{B,}X_B)$. However, nothing prevents Bob from claiming that his public key is $X'_B = X_B - X_A$.
- If he does so, others will assume that $X_A + X'_B$is the aggregated key that Alice and Bob need to cooperate in order to sign for
- Unfortunately, that is equal to X_B, thus Bob can clearly sign for this by himself
- This is called a rogue-key attack
- One way to avoid this is requiring that Alice and Bob prove first that they actually possess the private keys corresponding to their claimed public keys; however this is not always possible
- Ideally a scheme needs to be constructed whose security does not rely on out-of-band verification of the keys.

4 Bellare and Neven

Bellare-Neven (BN) is a more widely known plain public-key multi-signature scheme, that does not support key aggregation. It is possible to use BN multi-signatures where the individual keys are MuSig aggregates. BN multi-signature scheme is secure without such assumptions. Below are details:

- Call $L = H(X_{1,}X_{2...})$
- Each signer chooses a random nonce r_i and shares $R_i = r_iG$ with the other signers
- Call R the sum of the R_i points
- Each signer computes $s_i = r_i + H(L, X_{i,}R, m)x_i$
- The final signature is (R, s) where $s$is the sum of the s_i values
- Verification requires $sG = R + H(L, X_{1,}R, m)X_2 + ...$

Technically, BN has a pre-commit round, where the signers initially reveal $H(R_i)$to each other, prior to revealing the R_i points themselves. This step is a requirement in order to prove security under the DL assumption, but it can be dismisses if instead the OMDL assumption is accepted.

Furthermore, when an IAS is desired (where each signer has their own message), $L = H((X_1, m_{1)}, (X_2, m_2), ...)$ and $s_i = r_i + H(L, R, i)x_i$ is used for signing (and analogous for verification).

The resulting signature does not satisfy the normal Schnorr equation anymore, nor any other equation that can be written as a function of a combination of the public keys; the key aggregation property is lost in order to gain security in the plain public-key model.

This is where MuSig comes in. It recovers the *key aggregation property without losing security:*

- Call $L = H(X_1, X_2...)$
- Call X the sum of all $H(L, X_i)X_i$
- Each signer chooses a random nonce r_i, and shares $R_i = r_iG$ with the other signers
- Call R the sum of the R_i points
- Each signer computes $s_i = r_i + H(X, R, m)H(L, X_i)x_i$
- The final signature is (R, s) where $s$is the sum of the s_i values
- Verification again satisfies $sG = R + H(X, R, m)X$

So what was needed was to define X not as a simple sum of the individual public keys X_i,but as a sum of multiples of those keys, where the multiplication factor depends on a hash of all participating keys. [3]

5 Simple Schnorr Multi-Signatures with Applications to Bitcoin

The paper describes a new Schnorr-based multi-signature scheme called MuSig, which is provably secure in the *plain public-model*. This means that signers are only required to have a public key, but they do not have to prove knowledge of the private key corresponding to their public key to some certification authority or to other signers prior to engaging the protocol.

This new scheme provides improvements to Bellare and Neven (ACM-CCS 2006) and its variants by Bagherzandi *et al.* (ACM-CCS 2008) and Ma *et al.* (Des. Codes Cryptogr., 2010) in two respects:

Write Sections on Bagerzandi and Ma

1. It is simple and efficient, as it has the same key and signature size as standard Schnorr signatures;

2. It allows *key aggregation*, where the joint signature can be verified just as a standard Schnorr signature with respect to a single "aggregated" public key which can be computed from the individual public keys of the signers. [1]

5.1 Multi-signatures

Introduced by Itakura and Nakamura [4], multi-signature protocols allow a group of signers (that individually possess their own private/public key pair) to produce a single signature σon a message m. Verification of the given signature σcan be publicly performed given the message and the set of public keys of all signers.

A simple way to change a standard signature scheme into a multi-signature scheme is to have each signer produce a stand-alone signature for m with its private key and to then concatenate all individual signatures.

The transformation of a standard signature scheme to a multi-signature scheme needs to useful and practical, thus the newly calculated multi-signature scheme must produce signatures where the size is independent of the number of signers and similar to that of the original signaute scheme. [1]

5.1.1 Rogue Attacks

Rogue attacks are a significant concern when implementing multi-signature schemes. Here a subset of corrupted singers, manipulate the public keys computed as functions of the public keys of honest users, allowing them to easily produce forgeries for the set of public keys (despite them not knowing the associated secret keys.

Proposals from [5], [6], [7], [8], [9], [10], [11] were thus undone before a formal model was put forward along with a provably secure scheme from Micali, Ohta, and Reyzin. [12]Unfortunately, despite being provably secure this scheme is costly and an impractical interactive key generation protocol. [1]

A means of generically preventing rogue-key attacks is to make it mandatory for users to prove knowledge (or possession [13]) of the secret key during public key registration with a certification authority. Certification authority is a setting known as the knowledge of secret key (KOSK) assumption. The pairing-based multi-signature schemes by Boldyreva [14] and Lu *et al.* [15] rely on the KOSK assumption in order to maintain security. However, this as can be seen from [16]and [13] this assumption is problematic.

As it stands, the Bellare and Neven [16] provides the most practical multi-signature scheme, based on the Schnorr signature scheme, which is provably secure that does not contain any assumption on the key setup. Since the only requirement of this scheme is that each potential signer has a public key, this setting is referred to as the *plain-key model.*

Should Bellare and Neven reference be included in this statement

5.1.2 Schnorr Signature Scheme

The Schnorr signature scheme uses:[17]

- A cyclic group G of prime order p
- A generator $g$of G
- A hash function H
- A private/public key pair is a pair $(x, X)\epsilon\{0, ..., p-1\}\times G$ where $X = g^x$
- To sign a message m, the signer draws a random integer $r$in Z_p, computes $R = g^r$, $c = H(X, R, m)$, and $s = r + cx$
- The signature is the pair (R, s) , and its validity can be checked by verifying whether $g^s = RX^c$

The above described is referred to as the so-called "key-prefixed" variant of the scheme, which sees the public key hashed together with R and m[18]. This variant was thought to have a better multi-user security bound than the classic variant [19], however in [20] the key-prefixing was seen as unnecessary to enable good multi-user security for Schnorr signatures.

For the development of the new Schnorr-based multi-signature scheme [1], key-prefixing seemed a requirement for the security proof to go through, despite not knowing the form of an attack. The rationale also follows the process in reality, as messages signed in Bitcoin always indirectly commits to the public key.

5.1.3 Design of a Schnorr multi-signature scheme

The naive way to design a Schnorr multi-signature scheme would be as follows:

- A group of $n$signers want to cosign a message m
- Let $L = \{X_1 = g^{x_1}, ..., X_n = g^{x_n}\}$ be the multi-set[1]

[1]No constraints are imposed on the key setup, the adversary thus can choose corrupted public keys at random, hence the same public key can appear more than once in L

References

[1] **G. Maxwell, A. Poelstra, Y. Seurin, and P. Wuille, "Simple Schnorr Multi-Signatures with Applications to Bitcoin," pp. 1–34, 2018.**

[2] **Blockstream, "Schnorr Signatures for Bitcoin - BPASE '18," 2018. [Online]. Available: https://blockstream.com/2018/02/15/schnorr-signatures-bpase/**

[3] **P. Wuille, "Key Aggregation for Schnorr Signatures," 2018. [Online]. Available: https://blockstream.com/2018/01/23/musig-key-aggregation-schnorr-signatures/**

[4] **K. Itakura, "A public-key cryptosystem suitable for digital multisignatures," *NEC J. Res. Dev.*, vol. 71, 1983.**

[5] **C.-M. Li, T. Hwang, and N.-Y. Lee, "Threshold-multisignature schemes where suspected forgery implies traceability of adversarial shareholders," in *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer, 1994, pp. 194–204.**

[6] **L. Harn, "Group-oriented (t, n) threshold digital signature scheme and digital multisignature," *IEE Proceedings-Computers and Digital Techniques*, vol. 141, no. 5, pp. 307–313, 1994.**

[7] **P. Horster, M. Michels, and H. Petersen, "Meta-Multisignature schemes based on the discrete logarithm problem," in *Information Securityâthe Next Decade*. Springer, 1995, pp. 128–142.**

[8] **K. Ohta and T. Okamoto, "A digital multisignature scheme based on the Fiat-Shamir scheme," in *International Conference on the Theory and Application of Cryptology*. Springer, 1991, pp. 139–148.**

[9] **S. K. Langford, "Weaknesses in some threshold cryptosystems," in *Annual International Cryptology Conference*. Springer, 1996, pp. 74–82.**

[10] **M. Michels and P. Horster, "On the risk of disruption in several multiparty signature schemes," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 1996, pp. 334–345.**

[11] K. Ohta and T. Okamoto, "Multi-signature schemes secure against active insider attacks," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 82, no. 1, pp. 21–31, 1999.

[12] S. Micali, K. Ohta, and L. Reyzin, "Accountable-subgroup multisignatures," in *Proceedings of the 8th ACM conference on Computer and Communications Security*. ACM, 2001, pp. 245–254.

[13] T. Ristenpart and S. Yilek, "The power of proofs-of-possession: Securing multiparty signatures against rogue-key attacks," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2007, pp. 228–245.

[14] A. Boldyreva, "Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme," in *International Workshop on Public Key Cryptography*. Springer, 2003, pp. 31–46.

[15] S. Lu, R. Ostrovsky, A. Sahai, H. Shacham, and B. Waters, "Sequential aggregate signatures and multisignatures without random oracles," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2006, pp. 465–485.

[16] M. Bellare and G. Neven, "Multi-Signatures in the Plain Public-Key Model and a General Forking Lemma," *Acm Ccs*, pp. 390–399, 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1180453

[17] C.-P. Schnorr, "Efficient signature generation by smart cards," *Journal of cryptology*, vol. 4, no. 3, pp. 161–174, 1991.

[18] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang, "High-speed high-security signatures," *Journal of Cryptographic Engineering*, vol. 2, no. 2, pp. 77–89, 2012.

[19] D. J. Bernstein, "Multi-user Schnorr security, revisited." *IACR Cryptology ePrint Archive*, vol. 2015, p. 996, 2015.

[20] E. Kiltz, D. Masny, and J. Pan, "Optimal security proofs for signatures from identification schemes," in *Annual Cryptology Conference*. Springer, 2016, pp. 33–61.