Unforgeability of MuSig2 with tweaking

No Author Given

No Institute Given

1 Modified Scheme

See Figure 2.

$\mathsf{Setup}(1^\lambda)$

$(\mathbb{G}, p, g) \leftarrow \mathsf{GrGen}(1^\lambda)$
Select three hash functions
$\mathsf{H}_{\mathrm{agg}}, \mathsf{H}_{\mathrm{non}}, \mathsf{H}_{\mathrm{sig}} : \{0,1\}^* \to \mathbb{Z}_p$
$par := ((\mathbb{G}, p, g), \mathsf{H}_{\mathrm{agg}}, \mathsf{H}_{\mathrm{non}}, \mathsf{H}_{\mathrm{sig}})$
return par

$\mathsf{KeyGen}()$

$x \leftarrow\!\$\ \mathbb{Z}_p$; $X := g^x$
$sk := x$; $pk := X$
return (sk, pk)

$\mathsf{KeyAggCoef}(L, X_i)$

return $\mathsf{H}_{\mathrm{agg}}(L, X_i)$

$\mathsf{KeyAgg}(L)$

$\{X_1, \ldots, X_n\} := L$
for $i := 1 \ldots n$ **do**
  $a_i := \mathsf{KeyAggCoef}(L, X_i)$
return $\widetilde{X} := \prod_{i=1}^n X_i^{a_i}$

$\mathsf{Ver}(\widetilde{pk}, m, \sigma)$

$\widetilde{X} := \widetilde{pk}$; $(R, s) := \sigma$
$c := \mathsf{H}_{\mathrm{sig}}(\widetilde{X}, R, m)$
return $(g^s = R\widetilde{X}^c)$

$\mathsf{Sign}()$

// Local signer has index 1.
for $j := 1 \ldots \nu$ **do**
  $r_{1,j} \leftarrow\!\$\ \mathbb{Z}_p$; $R_{1,j} := g^{r_{1,j}}$
$out_1 := (R_{1,1}, \ldots, R_{1,\nu})$
$state_1 := (r_{1,1}, \ldots, r_{1,\nu})$
return $(out_1, state_1)$

$\mathsf{SignAgg}(out_1, \ldots, out_n)$

for $i := 1 \ldots n$ **do**
  $(R_{i,1}, \ldots, R_{i,\nu}) := out_i$
for $j := 1 \ldots \nu$ **do**
  $R_j := \prod_{i=1}^n R_{i,j}$
return $out := (R_1, \ldots, R_\nu)$

$\mathsf{Sign}'(state_1, out, sk_1, m, (pk_2, \ldots, pk_n), C)$

// Sign' must be called at most once per $state_1$.
$(r_{1,1}, \ldots, r_{1,\nu}) := state_1$
$x_1 := sk_1$; $X_1' := g^{x_1}$; $t := \mathsf{H}_{\mathrm{contract}}(X_1', C)$; $X_1 := X_1' g^t$
$(X_2, \ldots, X_n) := (pk_2, \ldots, pk_n)$
$L := \{X_1, \ldots, X_n\}$
$a_1 := \mathsf{KeyAggCoef}(L, X_1)$
$\widetilde{X} := \mathsf{KeyAgg}(L)$
$(R_1, \ldots, R_\nu) := out$
$b := \mathsf{H}_{\mathrm{non}}(\widetilde{X}, (R_1, \ldots, R_\nu), m)$
$R := \prod_{j=1}^\nu R_j^{b^{j-1}}$
$c := \mathsf{H}_{\mathrm{sig}}(\widetilde{X}, R, m)$
$s_1 := ca_1(x_1 + t) + \sum_{i=1}^\nu r_{1,j} b^{j-1} \bmod p$
$state_1' := R$; $out_1' := s_1$
return $(state_1', out_1')$

$\mathsf{SignAgg}'(out_1', \ldots, out_n')$

$(s_1, \ldots, s_n) := (out_1', \ldots, out_n')$
$s := \sum_{i=1}^n s_i \bmod p$
return $out' := s$

$\mathsf{Sign}''(state_1', out')$

$R := state_1'$; $s := out'$
return $\sigma := (R, s)$

Fig. 1. The modified multi-signature scheme $\mathsf{MuSig2Tweak}[\mathsf{GrGen}, \nu]$. Modifications in red.

2 Wagner

The attack relies on Wagner's algorithm for solving the Generalized Birthday Problem, which can be defined as follows for the purpose of this paper: Given a constant value $t \in \mathbb{Z}_p$, an integer $k_{\max}$, and access to random oracle H mapping onto $\mathbb{Z}_p$, find a set $\{q_1, \ldots, q_{k_{\max}}\}$ of $k_{\max}$ queries such that $\sum_{k=1}^{k_{\max}} H(q_k) = t$. For $k_{\max} = 2^{\sqrt{\log_2(p)}-1}$ the complexity of this algorithm is $O(2^{2\sqrt{\log_2(p)}})$.

Jonas' note: Perhaps can use BLOR? "If the attacker is able to open more sessions concurrently, the improved polynomial-time attack by Benhamouda *et al.* [**add:BLOR20**] assumes $k_{\max} > \log_2 p$ sessions, but then has complexity $O(k_{\max} \log_2 p)$ and a negligible running time in practice."

3 Attack

The attack proceeds as follows. The adversary opens $k_{\max} = 2^{\sqrt{\log_2(p)}-1}$ concurrent signing sessions and receives $k_{\max}$ nonce pairs $R_{1,1}^{(1)}, R_{1,2}^{(1)}, \ldots, R_{1,1}^{(k_{\max})}, R_{1,2}^{(k_{\max})}$ from the honest signer with public key $X_1 = g^{x_1}$.

The adversary draws $\ell_{\max} \in O(2^{2\sqrt{\log_2(p)}})$ values $C_1, \ldots, C_{\ell_{\max}}$ at random. Let $L = \{X_1 g^{t^{(1)}}, \ldots, X_1 g^{t^{(\ell_{\max})}}\}$ where $t^{(\ell)}$ are the tweaks and $\widetilde{X} = KeyAgg(L)$ be the corresponding aggregate public key.

Jonas' note: The attack probably doesn't require such large L but can work with multiple smaller multisets of keys? NO if we'd select between different L then b would change between the sessions which changes target R^*.

Given $R_1 = \sum_{k=1}^{k_{\max}} R_{1,1}^{(k)}, R_2 = \sum_{k=1}^{k_{\max}} R_{1,2}^{(k)}$ and a forgery target message m^*, the adversary computes $b = \mathsf{H}_{\text{non}}(\widetilde{X}, (R_1, R_2), m)$ and $R^* = \prod_{k=1}^{k_{\max}} R_{1,1}^{(k)} (R_{1,2}^{(k)})^b$. and uses Wagner's algorithm to find a $t^{(\ell)}, 1 \le \ell \le \ell_{\max}$ for each session $k, 1 \le k \le k_{\max}$ (in other words a function $f : [1, k_{\max}] \to [1, \ell_{\max}]$) such that

$$\sum_{k=1}^{k_{\max}} \underbrace{\mathsf{H}_{\text{agg}}(L, X_1 g^{t^{(f(k))}})}_{=: a_1^{(k)}} \underbrace{\mathsf{H}_{\text{sig}}(\widetilde{X}, R^*, m)}_{=: c^{(k)}} = \underbrace{\mathsf{H}_{\text{sig}}(X_1, R^*, m^*)}_{=: c^*}. \tag{1}$$

The honest signer will reply with $k_{\max}$ partial signatures $s_1^{(k)} = r_{1,1}^{(k)} + b r_{1,2}^{(k)} + c^{(k)} \cdot a_1^{(k)} (x_1 + t^{(f(k))})$. The adversary is able to obtain

$$s_1^{*\prime} = \sum_{k=1}^{k_{\max}} s_1^{(k)} \tag{2}$$

$$= \sum_{k=1}^{k_{\max}} \left(r_{1,1}^{(k)} b r_{1,2}^{(k)} \right) + \left(\sum_{k=1}^{k_{\max}} c^{(k)} a_1^{(k)} \right) \cdot x_1 + \sum_{k=1}^{k_{\max}} c^{(k)} a_1^{(k)} t^{(f(k))} \tag{3}$$

$$= \log_g(R^*) + c^* x_1 + \sum_{k=1}^{k_{\max}} c^{(k)} a_1^{(k)} t^{(f(k))} \tag{4}$$

where the last equality follows from Equation (1). Then the adversary can subtract the unwanted term as follows

$$s_1^* = s_1^{*\prime} - \sum_{k=1}^{k_{\max}} c^{(k)} a_1^{(k)} t^{(f(k))}$$

to obtain (R^*, s^*), a valid forgery on message m^* for public key X_1.

4 (Preliminary) Conclusion

However, Equation (2) seems to indicate that if the honest signer signs with multiple random keys instead of tweaked keys then this attack does not apply. The adversary would only be able to obtain a $s_1^* = \log_g(R^*) + \sum_{k=1}^{k_{\max}} c^{(k)} a_1^{(k)} x^{(f(k))}$.

5 Possible Fix

See Figure 2.

$\mathsf{Setup}(1^\lambda)$

$(\mathbb{G}, p, g) \leftarrow \mathsf{GrGen}(1^\lambda)$
Select three hash functions
$\mathsf{H}_{\text{agg}}, \mathsf{H}_{\text{non}}, \mathsf{H}_{\text{sig}} : \{0,1\}^* \to \mathbb{Z}_p$
$par := ((\mathbb{G}, p, g), \mathsf{H}_{\text{agg}}, \mathsf{H}_{\text{non}}, \mathsf{H}_{\text{sig}})$
$T_{\text{sess}} := \emptyset$
return par

$\mathsf{KeyGen}()$

$x \leftarrow\!\$\ \mathbb{Z}_p$; $X := g^x$
$sk := x$; $pk := X$
return (sk, pk)

$\mathsf{KeyAggCoef}(L, X_i)$

return $\mathsf{H}_{\text{agg}}(L, X_i)$

$\mathsf{KeyAgg}(L)$

$\{X_1, \ldots, X_n\} := L$
for $i := 1 \ldots n$ **do**
  $a_i := \mathsf{KeyAggCoef}(L, X_i)$
return $\widetilde{X} := \prod_{i=1}^n X_i^{a_i}$

$\mathsf{Ver}(\widetilde{pk}, m, \sigma)$

$\widetilde{X} := \widetilde{pk}$; $(R, s) := \sigma$
$c := \mathsf{H}_{\text{sig}}(\widetilde{X}, R, m)$
return $(g^s = R\widetilde{X}^c)$

$\mathsf{Sign}(sk_1, t)$

// Local signer has index 1.
for $j := 1 \ldots \nu$ **do**
  $r_{1,j} \leftarrow\!\$\ \mathbb{Z}_p$; $R_{1,j} := g^{r_{1,j}}$
$out_1 := (R_{1,1}, \ldots, R_{1,\nu})$
$state_1 := (r_{1,1}, \ldots, r_{1,\nu})$
$T_{\text{sess}}(r_{1,1}, \ldots, r_{1,\nu}) = sk_1 + t \bmod p$
return $(out_1, state_1)$

$\mathsf{SignAgg}(out_1, \ldots, out_n)$

for $i := 1 \ldots n$ **do**
  $(R_{i,1}, \ldots, R_{i,\nu}) := out_i$
for $j := 1 \ldots \nu$ **do**
  $R_j := \prod_{i=1}^n R_{i,j}$
return $out := (R_1, \ldots, R_\nu)$

$\mathsf{Sign}'(state_1, out, m, (pk_2, \ldots, pk_n))$

// Sign' must be called at most once per $state_1$.
$(r_{1,1}, \ldots, r_{1,\nu}) := state_1$
$x_1 := T_{\text{sess}}(r_{1,1}, \ldots, r_{1,\nu})$
$X_1 := g^{x_1}$
$(X_2, \ldots, X_n) := (pk_2, \ldots, pk_n)$
$L := \{X_1, \ldots, X_n\}$
$a_1 := \mathsf{KeyAggCoef}(L, X_1)$
$\widetilde{X} := \mathsf{KeyAgg}(L)$
$(R_1, \ldots, R_\nu) := out$
$b := \mathsf{H}_{\text{non}}(\widetilde{X}, (R_1, \ldots, R_\nu), m)$
$R := \prod_{j=1}^\nu R_j^{b^{j-1}}$
$c := \mathsf{H}_{\text{sig}}(\widetilde{X}, R, m)$
$s_1 := ca_1x_1 + \sum_{i=1}^\nu r_{1,j} b^{j-1} \bmod p$
$state_1' := R$; $out_1' := s_1$
return $(state_1', out_1')$

$\mathsf{SignAgg}'(out_1', \ldots, out_n')$

$(s_1, \ldots, s_n) := (out_1', \ldots, out_n')$
$s := \sum_{i=1}^n s_i \bmod p$
return $out' := s$

$\mathsf{Sign}''(state_1', out')$

$R := state_1'$; $s := out'$
return $\sigma := (R, s)$

Fig. 2. The modified multi-signature scheme $\mathsf{MuSig2Tweak}[\mathsf{GrGen}, \nu]$. Modifications in red.