Security of Hedged Fiat–Shamir Signatures under Fault Attacks

Diego F. Aranha, Claudio Orlandi, Akira Takahashi, and Greg Zaverucha

ambiso

2020-05-06

Introduction

- Consider security of hedged signature schemes against fault attacks
- Analysis and model is specific to Fiat-Shamir transformed identification protocols

Hedged Signatures

Nonce explicitly given to sign function

HSign(sk, m, n)

Hedged extractor (= hash function modeled as ROM)[1]

$\rho \leftarrow \mathsf{HE}(sk, (m, n))$

$\sigma \leftarrow \mathsf{Sign}(sk, m; \rho)$

$\textbf{return}\ \sigma$

[1] instantiated with PRF

Game Based Security Definitions

Introduce new security notions for chosen nonces and faulting:

- UF-KOA *(unforgeability under key only attack)*
- UF-CMA *(unforgeability under chosen message attack)*

* UF-CMNA *(unforgeability under chosen message and nonce attack)*
* F-UF-fCMA *(unforgeability under faults, chosen message attack)*
* F-UF-fCMNA *(unforgeability under faults, chosen message and nonce attack)*
  ↑ Set of fault types/positions

UF-CMA $\xrightarrow{R2H}$ UF-CMNA $\longrightarrow$ F-UF-fCMNA
↑ For specific F

UF-CMA (unforgeability under chosen message attack)

$\mathsf{Exp}^{\mathsf{UF\text{-}CMA}}_{\mathsf{SIG}}(\mathcal{A})$	$\mathsf{OSign}(m)$	$\mathsf{H}(x)$
$M \leftarrow \emptyset; \mathrm{HT} \leftarrow \emptyset$	$\rho \leftarrow_{\$} D_\rho$	If $\mathrm{HT}[x] = \bot$:
$(sk, pk) \leftarrow \mathsf{Gen}(1^\lambda)$	$\sigma \leftarrow \mathsf{Sign}(sk, m; \rho)$	$\quad \mathrm{HT}[x] \leftarrow_{\$} D_H$
$(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{OSign},\mathsf{H}}(pk)$	$M \leftarrow M \cup \{m\}$	**return** $\mathrm{HT}[x]$
$v \leftarrow \mathsf{Verify}(pk, m^*, \sigma^*)$	**return** σ	
return $(v = 1) \wedge m^* \notin M$		

Figure 2: Standard UF-CMA experiment in the random oracle model

Figure from [Ara+19].

UF-CMNA (unforgeability under chosen message and nonce attack)

$\mathsf{HSign}(sk, m, n)$

$\rho \leftarrow \mathsf{HE}(sk, (m, n))$

$\sigma \leftarrow \mathsf{Sign}(sk, m; \rho)$

return σ

$\mathsf{Exp}^{\mathsf{UF\text{-}CMNA}}_{\mathsf{HSIG,HE}}(\mathcal{A})$

$M \leftarrow \emptyset; \mathrm{HET} \leftarrow \emptyset$

$(sk, pk) \leftarrow \mathsf{Gen}(1^\lambda)$

$(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{OHSign,HE}}(pk)$

$v \leftarrow \mathsf{Verify}(m^*, \sigma^*)$

return $(v = 1) \wedge m^* \notin M$

$\mathsf{OHSign}(m, n)$

$\sigma \leftarrow \mathsf{HSign}(sk, m, n)$

$M \leftarrow M \cup \{m\}$

return σ

$\mathsf{HE}(sk', (m', n'))$

If $\mathrm{HET}[sk', m', n'] = \perp$:

  $\mathrm{HET}[sk', m', n'] \leftarrow_\$ D_\rho$

return $\mathrm{HET}[sk', m', n']$

Figure 5: Hedged signature scheme $\mathsf{HSIG} = \mathbf{R2H}[\mathsf{SIG}, \mathsf{HE}] = (\mathsf{Gen}, \mathsf{HSign}, \mathsf{Verify})$ and UF-CMNA experiment. Key generation and verification are unchanged.

Figure from [Ara+19].

Introducing: Faults

- Only single bit transient faults considered
- The surveyed practical attacks can be performed with a single bit fault
- Faulting functions:

  $\texttt{flip\_bit}_i(x)$: Flips bit at position i

  $\texttt{set\_bit}_{i,b}(x)$: Sets bit at position i to b

  $Id(x)$: Identity function
- Adversary can choose where to apply which function

F-UF-fCMNA (unforgeability under faults, chosen message and nonce attack)

$\mathsf{Exp}^{\mathsf{UF\text{-}fCMA}}_{\mathsf{FS}}(\mathcal{A})$ | **$\mathsf{Exp}^{\mathsf{UF\text{-}fCMNA}}_{\mathsf{HFS,HE}}(\mathcal{A})$**

$M \leftarrow \emptyset; \mathrm{HT} \leftarrow \emptyset;$ $\boxed{\mathrm{HET} \leftarrow \emptyset}$

$(sk, pk) \leftarrow \mathsf{Gen}(1^\lambda)$

$(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{OFaultSign,H}}(pk)$

$\boxed{(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{OFaultHSign,H,HE}}(pk)}$

$v \leftarrow \mathsf{Verify}(m^*, \sigma^*)$

return $(v = 1) \wedge m^* \notin M$

$\mathsf{OFaultHSign}(m, n, j, \phi)$

$f_j := \phi; f_k := Id$ for $k \neq j$

FaultHSign

$\rho \leftarrow f_2(\mathsf{HE}(f_1(sk), f_0(m, n)))$

$(a, St) \leftarrow f_4(\mathsf{Com}(f_3(sk; \rho)))$

$\hat{a}, \hat{m}, \hat{pk} \leftarrow f_5(a, m, pk)$

$e \leftarrow f_6(\mathsf{H}(\hat{a}, \hat{m}, \hat{pk}))$

$z \leftarrow f_8(\mathsf{Resp}(f_7(sk, e, St)))$

$\sigma \leftarrow f_{10}(\mathsf{CSF}(f_9(a, e, z)))$

$M \leftarrow M \cup \{\hat{m}\}$

return σ

$\mathsf{OFaultSign}(m, j, \phi)$

$f_j := \phi; f_k := Id$ for $k \neq j$

FaultSign

$\rho \leftarrow_\$ D_\rho; \rho \leftarrow f_2(\rho)$

$(a, St) \leftarrow f_4(\mathsf{Com}(f_3(sk; \rho)))$

$\hat{a}, \hat{m}, \hat{pk} \leftarrow f_5(a, m, pk)$

$e \leftarrow f_6(\mathsf{H}(\hat{a}, \hat{m}, \hat{pk}))$

$z \leftarrow f_8(\mathsf{Resp}(f_7(sk, e, St)))$

$\sigma \leftarrow f_{10}(\mathsf{CSF}(f_9(a, e, z)))$

$M \leftarrow M \cup \{\hat{m}\}$

return σ

Figure 6: UF-fCMNA and UF-fCMA security experiments and faulty signing oracles for both hedged (HFS) and plain (FS) Fiat–Shamir signature schemes. Id stands for the identity function. The function H and HE (not shown), are the same as in Fig. 4 and Fig. 5, respectively. A dashed box indicates that the instructions inside correspond to the actual faulty signing operation.

Figure from [Ara+19].

Trivial Attacks

Faulting the message in the first step $\rightarrow$ same as simply querying a different message

- Doesn't count as an actual attack
- Excluded from the model by recording faulted messages

Comparison to [FG20]

Differences in the adversarial power as noted by [Ara+19].

In [Ara+19]'s model the adversary can:

Comparison to [FG20]

Differences in the adversarial power as noted by [Ara+19].

In [Ara+19]'s model the adversary can:

* Set a bit to a chosen value
    - Full or differential faults cannot emulate $\texttt{set\_bit}_{i,b}$, since $\mathcal{A}$ would need to know the read value beforehand

Comparison to [FG20]

Differences in the adversarial power as noted by [Ara+19].

In [Ara+19]'s model the adversary can:

* Set a bit to a chosen value
    - Full or differential faults cannot emulate $\texttt{set\_bit}_{i,b}$, since $\mathcal{A}$ would need to know the read value beforehand
- Fault sk before hashing
    - Attack:
        - Generate $|sk|$ signatures
        - Use $\texttt{set\_bit}_{i,b}$ to set the i^{th}-bit to 0
        - If the signature verifies, the i^{th}-bit is 0 with high probability
    - sk was removed from [FG20]'s fault-resiliency-analysis: "[...] considering fault attacks on sk also in the signing process will require signature schemes secure against related-key attacks [...]" [FG20] (but this wouldn't hold with $\texttt{set\_bit}_{i,b}$ since the key can easily be extracted)

Comparison to [FG20] (continued . . .)

- Exercise "nearly full control over the nonce, instead of assuming nonces are randomly generated and subject to bit flips later on" [Ara+19]
  - But there doesn't seem to be any difference to whether you supply the nonce or later flip it whenever you want?

Comparison to [FG20] (continued . . .)

- Exercise "nearly full control over the nonce, instead of assuming nonces are randomly generated and subject to bit flips later on" [Ara+19]
  - But there doesn't seem to be any difference to whether you supply the nonce or later flip it whenever you want?
- Faulting of intermediate values (Fiat-Shamir specific) $\rightarrow$ more fine grained analysis
  - More fine grained analysis found that Picnic2 (signature scheme) is $\{f_2\}$-UF-fCMNA secure, but that this doesn't hold in general (opportunity for capturing new security notions)
  - Suggested mitigation to faulting the commitment in Picnic2 may depend on only a single fault being possible: "In the one-fault model, two signatures is always sufficient."

Comparison to [FG20] (continued . . .)

But:

- Only single bit fault inserted into a single value
  - Authors argue the surveyed practical attacks can be performed with a single bit fault

Comparison to [FG20] (continued . . .)

But:

- Only single bit fault inserted into a single value
  - Authors argue the surveyed practical attacks can be performed with a single bit fault
- Clash not (explicitly?) considered
  (if $\mathcal{A}$ obtains a signature that verifies with multiple messages)

Remarks

- Cluttered formalization
  - $\langle \cdot \rangle$ syntax from[FG20] seems quite convenient
  - Not obvious what value is faulted in e.g. $\{f_0\}$-UF-fCMNA insecure scheme: $f_0(m, n)$ (proof faults the message, but could we fault the nonce instead?)
  - Referring to specific fault positions by a unique identifier seems useful
  - Maybe combine the notation? E.g. $\langle m \rangle_1$ is the first occurrence where m is read.

Remarks (continued...)

- Remarks section has some interesting suggestions
  - Faulting global parameters
  - Faulting signature specific/internal computations isn't covered (e.g. faulting intermediate results of a sign operation)
  - Instruction skip attacks
    - Fault control flow
    - Likely implementation (and hardware) dependent
  - Fiat-Shamir *with aborts* (lattice crypto rejection sampling)

Summary

- Game based security notion
- Cluttered formalism
- Fine grained analysis $\rightarrow$ messy details
  But: potential for new signature security requirements
- Only single bit transient faults that occur once permitted
- Full faults and $\texttt{set\_bit}_{i,b}$ don't map to each other

References

[Ara+19] Diego F. Aranha, Claudio Orlandi, Akira Takahashi, and Greg Zaverucha. *Security of Hedged Fiat-Shamir Signatures under Fault Attacks*. Cryptology ePrint Archive, Report 2019/956. https://eprint.iacr.org/2019/956. 2019.

[FG20] Marc Fischlin and Felix Günther. "Modeling memory faults in signature and authenticated encryption schemes". In: *Cryptographers' Track at the RSA Conference*. Springer. 2020, pp. 56–84.