Proof Obligations Induced by Shared Challenges in Hybrid Fiat-Shamir Signatures

Sara Zain
Barkhausen Institut, Dresden, Germany

***Abstract*—The FS-FS hybrid signature scheme [1] combines two Fiat-Shamir components through a single shared challenge. Its EUF-CMA security is stated as a theorem in that work-in-progress, but no proof is provided. We report an EasyCrypt mechanisation of the FS-FS security argument in the Random Oracle Model (ROM), defined over abstract Fiat-Shamir components and instantiated with Schnorr, with ongoing work toward broader instantiations and the QROM. The formalisation yields two distinct insights. First, the second-preimage resistance assumption appearing in the original theorem statement is subsumed by the standard ROM guessing bound. Second, establishing the reduction in EasyCrypt requires two additional proof obligations: a logging invariant for the guessing branch and a tracker invariant for the collision-resistance reduction. Together, these obligations reveal proof-structural complexity introduced by the shared challenge that is absent from single-component Fiat-Shamir signatures.**

***Index Terms*—Hybrid signatures, Fiat-Shamir**

I. Introduction

Hybrid signature schemes combining a classical and a post-quantum component are under active standardisation in NIST and IETF working groups [2]. Their motivation is robustness during the transition period: if one component proves vulnerable, the other continues to provide security. Among the constructions proposed by Bindel and Hale [1], the FS-FS hybrid is notable for deriving a single shared challenge combining from two Fiat-Shamir [3] components, intended as a heterogeneous classical/post-quantum pair. This shared-challenge structure enables simultaneous verification and proof composability, but also introduces dependencies absent from ordinary Fiat-Shamir signatures. A successful forgery must therefore be analysed with respect to a challenge determined jointly by both components.

The original work is work-in-progress: Theorem 1 states EUF-CMA security but does not provide a complete proof [1]. This makes FS-FS a natural target for machine-checked analysis. Prior machine-checked proofs of single-component schemes—including Dilithium [4], SPHINCS+ [5], and Saber [6]—have each uncovered scheme-specific gaps. In FS-FS, the source of complexity is not a particular component scheme but the shared challenge linking both components.

We report an EasyCrypt mechanisation of the ROM security proof of the parametric FS-FS construction, instantiated with Schnorr [2]. For brevity, this workshop paper presents the Σ_1 reduction; the Σ_2 case is symmetric.

The mechanisation yields two main insights. First, the second-preimage-resistance (SPR) assumption appearing in the original theorem statement is subsumed by the standard ROM guessing bound and is therefore unnecessary. Second, establishing the reduction requires explicit proof obligations—a logging invariant, an auxiliary exposure argument, and a tracker-preservation argument—that arise from the shared challenge and remain implicit in the original informal proof.

II. Shared-Challenge Structure

EasyCrypt [7] is a proof assistant for game-based cryptographic security proofs [8] using the *game-hopping* methodology [8]: a security argument is a sequence of probabilistic programs, each step either preserving probability (equivalence) or bounding it (splitting on an event).

The FS-FS construction [1] (Alg. 10–11) combines two abstract FS components Σ_1 and Σ_2 (via `SigmaComp` interfaces in EasyCrypt), each with commitment P_i, response f_i, and reconstruction Rec_i; both share $\mathcal{H}$ and $\mathcal{D}$ but use independent identification protocols. Signing samples r_i, computes commitments $w_i = P_i(sk_i, r_i)$ (following [1], w_i is the commitment, i.e., first message of the Σ-protocol), then derives a *single shared challenge*:

$$x_V = (w_1,\, w_2,\, \mathcal{D}(m)), \quad c = \mathcal{H}(x_V), \quad z_i = f_i(c, r_i, sk_i).$$

The signature is (c, z_1, z_2); verification checks both component conditions together with $\mathcal{H}(\mathsf{Rec}_1(\cdot), \mathsf{Rec}_2(\cdot), \mathcal{D}(m)) = c$. The components may be heterogeneous: one classical (e.g. elliptic-curve FS) and one post-quantum (e.g. Dilithium-style lattice FS).[3]

From a ROM perspective, the value x_V is the point at which any successful forgery must interact with the oracle. A game-hopping argument splits adversarial success by

[1]The theorem states [1] that if one component is EUF-CMA secure in the ROM, $\mathcal{D}$ is collision resistant, and $\mathcal{H}$ is second-preimage resistant, then the hybrid scheme is EUF-CMA secure.

[2]The development is available at https://gitlab.barkhauseninstitut.org/szain/pqc-classic-project/-/blob/main/FS-hybrid.ec?ref_type=heads

[3]The EasyCrypt development parameterises over abstract `SigmaComp` interfaces (Schnorr: $w_i = g^{k_i}$, $z_i = k_i + c \cdot sk_i$). The three obligations arise from the shared-challenge structure of x_V, not this instantiation, and persist for any P_i, f_i.

two events—badGuess (the adversary never queried $\mathcal{H}$ at x_V) and badD (a collision under $\mathcal{D}$). The original Theorem-1 of [1] lists EUF-CMA of one component, CR of $\mathcal{D}$, and SPR of $\mathcal{H}$. Our formal development leads to the bound:

$$\Pr[\mathsf{EUF\text{-}CMA}_{\Sigma_h}] \;\le\; \frac{1}{|\mathcal{R}|} + \Pr[\mathsf{EUF\text{-}CMA}_{\Sigma_1}] + \Pr[\mathsf{CR}_{\mathcal{D}}]\,, \tag{1}$$

where $|\mathcal{R}|$ is the cardinality of the oracle's output range; in the EasyCrypt development, this equals the parameter q of the distribution dp (the challenge space size). SPR does not appear: modelling $\mathcal{H}$ in the ROM already implies it, a standard consequence of the idealisation rather than a weakening of Theorem 1's hypotheses.

III. Proof Obligations Induced by Coupling

The key challenge is that x_V depends on both components, forcing the proof to track oracle interactions across two signing paths simultaneously.

The proof proceeds by four game hops (G0 $\leftrightarrow$ G1 $\leftrightarrow$ G2 $\rightarrow$ G3; Fig. 1), splitting on badGuess and badD. The analysis yields one theorem-level observation (absorption of SPR) and two proof obligations that arise explicitly in the mechanised reduction. The bound in (1) reduces to EUF-CMA of Σ_1; a symmetric Σ_2 reduction follows by swapping component roles.

Absorbing SPR. The original theorem lists SPR of $\mathcal{H}$ as an explicit assumption. In the ROM, however, $\mathcal{H}$ is modelled as a random oracle, and the adversary's success without querying $\mathcal{H}(x_V)$ is already bounded by $1/|\mathcal{R}|$ via oracle uniformity. SPR is therefore subsumed by the standard ROM guessing argument and need not appear separately in the final bound.

This is a standard consequence of ROM-style Fiat-Shamir proofs, not an independent simplification: the asymmetry with $\mathcal{D}$–left abstract, hence requiring an explicit CR assumption–reflects a modelling choice rather than a property of the construction; idealising $\mathcal{D}$ analogously would absorb its CR hypothesis in the same way.

Logging invariant. To bound Pr[badGuess], one must show that the oracle output at (xV) is genuinely fresh, i.e., the adversary never queried $\mathcal{H}$ at (xV). In EasyCrypt, the random oracle is a lazy mutable map (`HRO.RO.m`; `HRO` is EasyCrypt's random oracle module for $\mathcal{H}$, `RO.m` its internal state). Freshness must be derived from program state via an invariant over a logging wrapper (`LH`) that records every $\mathcal{H}$-query:

$$\mathsf{Inv} \;:=\; \forall t.\; t \in \texttt{HRO.RO.m} \;\Rightarrow\; t \in \texttt{LH.qH}.$$

When badGuess holds, the shared challenge has not been queried to the random oracle and therefore corresponds to a fresh uniform sample. Establishing this requires the logging invariant (P1), while applying the oracle-uniformity argument requires auxiliary exposure (P2).

Tracker and restriction. The collision-resistance reduction must extract two distinct messages with the same digest. This requires a tracker invariant that records a valid collision witness whenever badD occurs, together with a module-restriction argument ensuring that subsequent oracle calls cannot modify the tracked state. Both obligations are implicit in the informal proof but must be established explicitly in EasyCrypt.

IV. Reusable Proof Patterns

The mechanisation exposes three recurring proof patterns for lazy-random-oracle ROM proofs.

P1–Logging invariant. For any "adversary never queried at x" branch: wrap $\mathcal{H}$ in a logging oracle LH; establish and preserve the invariant Inv (§III) across every oracle-touching module; derive $\texttt{HRO.RO.m}[x] = \texttt{None}$ at the branch point. Without this, the `byphoare` tactic—rule for deriving upper bounds on event probabilities via probabilistic Hoare logic(unlike `byequiv`, which handles exact probability equalities between two programs, `byphoare` handles inequalities on a single program)—cannot close the guessing bound. *Scope*: any RO freshness proof.

P2–Auxiliary exposure. Lazy lookup hides fresh sample from `byphoare`; auxiliary game saves forged challenge pre-lookup (event-equivalent). When a fresh uniform sample is buried inside a lazy map lookup, `byphoare` requires a deterministic post-condition on the sampled value. *Combines with P1*: P1 proves freshness, P2 exposes value. *Scope*: guessing bounds.

P3–Module restriction. When a reduction wraps an abstract module M in a tracker, use EasyCrypt's module restriction to enforce that M cannot write to the tracker's state. Discharge as an explicit lemma; informal proofs assume this silently. *Scope*: any reduction involving an abstract component module. Arises independently of P1 and P2.

V. Discussion

Two directions follow naturally. First, extending the development to heterogeneous instantiations would capture the intended classical/post-quantum hybrid setting. Second, extending the proof to the QROM [9] requires replacing the classical logging argument with quantum-compatible techniques such as compressed oracles [10] or one-way-to-hiding [11]. The original authors subsequently confirmed that the FS-FS proposal remained a work in progress and was not extended with a complete proof-oriented treatment. Our mechanisation therefore provides an independent clarification of its security argument.

VI. Conclusion

We identified three proof obligations induced by shared-challenge coupling and established the bound in (1). The obligations reside in the game-hopping infrastructure around x_V and are independent of the specific FS instantiation, persisting in any heterogeneous classical-plus-post-quantum deployment of the original framework.

References

[1] N. Bindel and B. Hale, "A note on hybrid signature schemes," *Cryptology ePrint Archive*, 2023.

[2] M. Ounsworth, J. Gray, and M. Pala, "Composite signatures for use in internet PKI," Internet-Draft, IETF, 2023. [Online]. Available: https://datatracker.ietf.org/doc/draft-ounsworth-pq-composite-sigs/

[3] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Conference on the theory and application of cryptographic techniques*. Springer, 1986, pp. 186–194.

[4] M. Barbosa, G. Barthe, C. Doczkal, J. Don, S. Fehr, B. Grégoire, Y.-H. Huang, A. Hülsing, Y. Lee, and X. Wu, "Fixing and mechanizing the security proof of Fiat-Shamir with aborts and Dilithium," in *Annual International Cryptology Conference*. Springer, 2023, pp. 358–389.

[5] M. Barbosa, F. Dupressoir, A. Hülsing, M. Meijers, and P.-Y. Strub, "A tight security proof for SPHINCS+, formally verified," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2024, pp. 35–67.

[6] A. Hülsing, M. Meijers, and P.-Y. Strub, "Formal verification of Saber's public-key encryption scheme in EasyCrypt," in *Annual International Cryptology Conference*. Springer, 2022, pp. 622–653.

[7] G. Barthe, F. Dupressoir, B. Grégoire, C. Kunz, B. Schmidt, and P.-Y. Strub, "Easycrypt: A tutorial," *International School on Foundations of Security Analysis and Design*, pp. 146–166, 2012.

[8] V. Shoup, "Sequences of games: a tool for taming complexity in security proofs," *cryptology eprint archive*, 2004.

[9] D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry, "Random oracles in a quantum world," in *International conference on the theory and application of cryptology and information security*. Springer, 2011, pp. 41–69.

[10] M. Zhandry, "How to record quantum queries, and applications to quantum indifferentiability," in *Annual International Cryptology Conference*. Springer, 2019, pp. 239–268.

[11] A. Ambainis, M. Hamburg, and D. Unruh, "Quantum security proofs using semi-classical oracles," in *Annual International Cryptology Conference*. Springer, 2019, pp. 269–295.

Appendix

We provide a high-level overview of the game-hopping structure used in the mechanised EasyCrypt development. The proof proceeds through a sequence of games and splits on the events badGuess and badD to isolate the guessing and collision branches. The Fig. 1 highlights where the three proof obligations identified in this work arise within the game sequence.

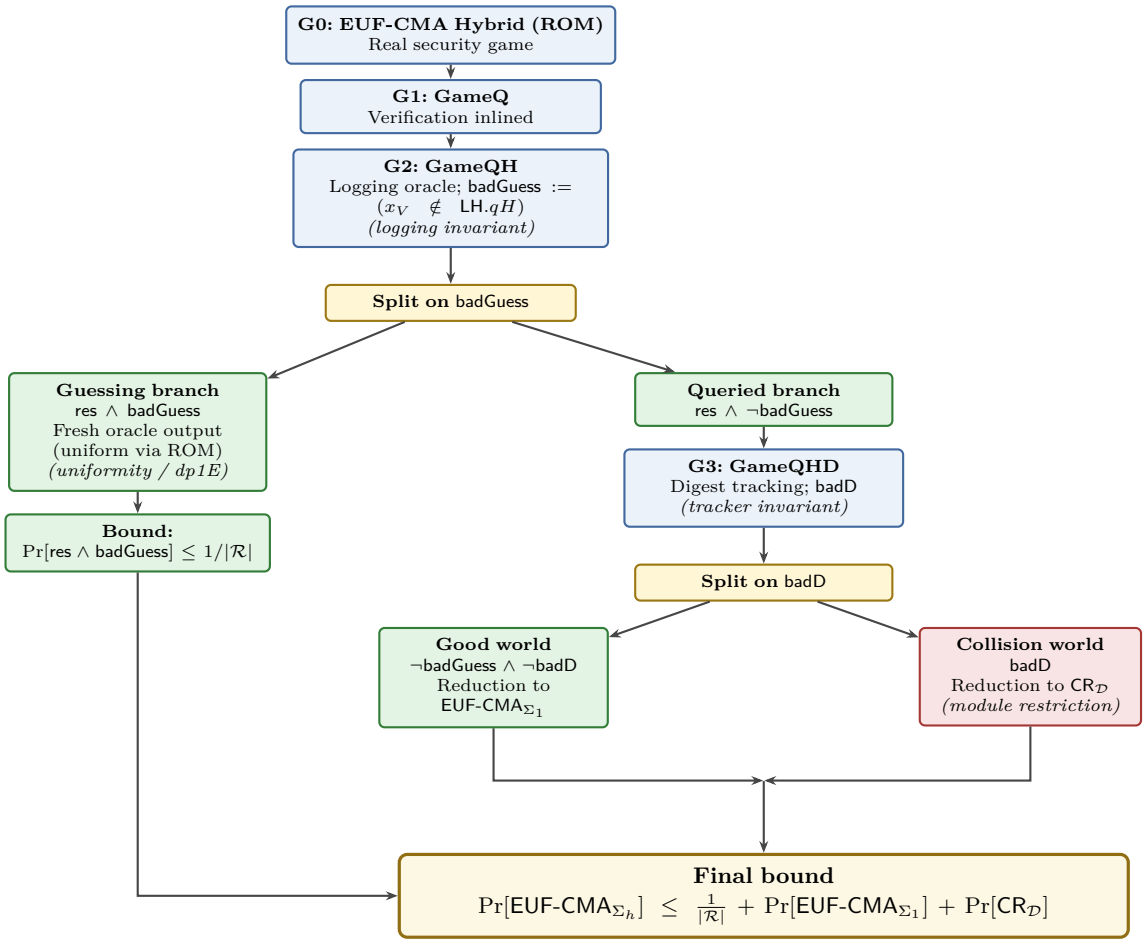

Fig. 1. Game-hopping structure of the mechanised ROM proof for the hybrid FS-FS signature scheme. G_0 is the original EUF-CMA experiment, G_1 inlines verification, G_2 augments the random oracle with query logging and introduces the event (badGuess), and G_3 introduces digest tracking and the collision event (badD). The transitions ($G_0 \leftrightarrow G_1 \leftrightarrow G_2$) are probability-preserving equivalences. Splitting on badGuess yields a guessing branch bounded by $1/|\mathcal{R}|$ and a queried branch that proceeds to G_3. Splitting on badD then separates the good world, which reduces to the EUF-CMA security of a component scheme (Σ_1), symmetrically (Σ_2), from the collision world, which reduces to the collision resistance of $\mathcal{D}$. Pattern (P1) (logging invariant) establishes oracle freshness in G_2 by relating the oracle table to the query log and is required to bound the badGuess branch. Pattern (P2) (auxiliary exposure) makes the freshly sampled challenge available to probabilistic reasoning, enabling the uniformity argument used in the $1/|\mathcal{R}|$ bound. Pattern (P3) (module restriction) preserves the digest-tracker state in G_3 and is required for the reduction from badD to collision resistance. All three arise from the shared challenge input $x_V = (w_1, w_2, \mathcal{D}(m))$, which couples the two Fiat-Shamir components at a single oracle query point. Combining the branches yields $\Pr[\text{EUF-CMA}_{\Sigma_h}] \leq \frac{1}{|\mathcal{R}|} + \Pr[\text{EUF-CMA}_{\Sigma_1}] + \Pr[\text{CR}_D]$.