Fiat-Shamir Goes Tropical

Rémi Géraud-Stewart[1][0000−0001−8719−1724], David Naccache[2], and Ofer Yifrach-Stav[2]

[1] Qualcomm Inc., San Diego, USA
[2] Département d'informatique de l'ENS, École normale supérieure, CNRS, PSL Research University, Paris, France
remi.geraud@ens.fr, david.naccache@ens.fr, ofer.friedman@ens.fr

Abstract. In a recent ePrint, Brown and Monico propose new attacks on the tropical signature scheme of Chen, Grigoriev and Shpilrain. This note provides a new countermeasure against *those* attacks. Thereby, we (temporarily?) shift the fire from the signature algorithm to redirect attacks on the key and on tropical polynomial factorization.

1 Introduction

This paper can be read a continuation of a thread of papers on a tropical signature scheme proposed by Chen, Grigoriev and Shpilrain (CGS) [2]. We refer the readers to [2,5,7], for the previous episodes of this saga. Further useful references on the topic are given in [6] and its bibliography.

In essence, the concerned signature scheme works as follows:

Denote by $\mathcal{P}_{r,d}$ the subset of $S[t]$ (the tropical polynomial semiring) with coefficients in $[0, r]$ and degree d.

Key generation: The signer selects $X, Y \in_R \mathcal{P}_{r,d}$. The public key is:

$$(r, d, M = X \otimes Y)$$

Signature: To sign m, the signer hashes m into $\mathrm{hash}(m) = H \in \mathcal{P}_{r,d}$.

The signer picks $U, V \in_R \mathcal{P}_{r,d}$ and computes the signature:

$$\sigma = (H, A, B, N) = (H, H \otimes X \otimes U, H \otimes Y \otimes V, U \otimes V)$$

Verification: To check σ, ascertain that:

- V1: $\mathrm{hash}(m) \stackrel{?}{=} H \in \mathcal{P}_{r,d}$
- V2: $A, B \stackrel{?}{\in} \mathcal{P}_{3r,3d}$
- V3: $N \stackrel{?}{\in} \mathcal{P}_{2r,2d}$
- V4: Neither A nor B is a constant tropical multiple of $H \otimes M$ or $H \otimes N$.
- V5: $A \otimes B \stackrel{?}{=} H \otimes H \otimes M \otimes N$

The fixing strategy will consist in translating the Fiat-Shamir protocol into the tropical realm (first fix) and by applying the Fiat-Shamir Transform (FST) to CGS (second fix).

For the sake of convenience we will denote for any collection of objects L_i:

$$\vec{L} = (L_0, L_1, \dots, L_{\tau-1})$$

Any operation $\star$ between arrowed variables is to be understood component wise, e.g.

$$\vec{L} \star \vec{L'} = (L_0 \star L'_0, L_1 \star L'_1, \dots, L_{\tau-1} \star L'_{\tau-1})$$

We define an auxiliary selection function for $L_0, L_1 \in \mathcal{P}_{\star,\star}$ and $b \in \{0,1\}$: $\Delta_b(L_0, L_1) = L_b$.

2 Tropicalized Fiat-Shamir

The first fix is interesting in that it translates directly a classical factoring-based scheme into a similar (hopefully) post-quantum scheme.

We use here the standard notations of [3]. The problem with the classical Fiat-Shamir is the fact that $s^2 v = 1 \bmod n$ which means that during key generation one needs to perform a modular division.

It is easy to work around this limitation by defining $s^2 = v \bmod n$ which results in the following scheme:

- The prover starts by picking randomly an $r \in_R \mathbb{Z}$;
- The prover sends a commitment $x = r^2 \bmod n$;
- The verifier replies with a challenge bit b;
- The prover responds with $y = s^b r \bmod n$;
- The verifier checks that $y^2 \stackrel{?}{=} v^b x \bmod n$.

To tropicalize this scheme, the secret key will become $S \in_R \mathcal{P}_{r,d}$ and the public key $V = S \otimes S \in \mathcal{P}_{2r,2d}$.

- The prover starts by picking randomly an $R \in_R \mathcal{P}_{r,d}$;
- The prover sends a commitment $X = R \otimes R$;
- The verifier replies with a challenge bit b;
- The prover responds with $Y = \Delta_b(R, S \otimes R)$;
- The verifier checks if $Y \otimes Y \stackrel{?}{=} \Delta_b(X, V \otimes X)$.

The verifier will also check that $V, X \stackrel{?}{\in} \mathcal{P}_{2r,2d}$ and $Y \stackrel{?}{\in} \mathcal{P}_{(1+b)r,(1+b)d}$.

To get a signature scheme from this zero-knowledge protocol one can just apply the Fiat-Shamir transform.

We note that the Chen, Grigoriev and Shpilrain differs from the above protocol in two points: the first is that it corresponds to a tropicalized Fiat-Shamir

where the challenge b is alwasys stuck to 1. The second is that squares are not used but the operation r^2 is replaced by $r_1 r_2$.

We do not know how to simulate the tropicalized Fiat-Shamir for the following reason. Following the traditional *modus operandi* the case $b = 0$ is trivial. For $b = 1$ we would use $X = R \otimes R \otimes V$ and $Y = R \otimes V$:

$$Y \otimes Y \stackrel{?}{=} V \otimes X$$

Indeed:

$$(R \otimes V) \otimes (R \otimes V) = V \otimes (R \otimes R \otimes V)$$

However, now $X \in \mathcal{P}_{4r,4d}$ and $Y \in \mathcal{P}_{3r,3d}$ which violates the verification conditions. This does not mean that the tropicalized Fiat-Shamir version is insecure but only that, do date, we don't know how to simulate it to prove its zero-knwoledgeness.

A potential way to get around this problem might be to increase in the legitimate protocol specifications to $R \in \mathcal{P}_{2r,2d}$. In which case the simulator could use "shorter than normal" Rs and the situation will be:

Table 1. Protocol modification to accommodate simulation.

when the challenge is $b = 1$	S	V	R	X	Y
legitimate protocol	$\mathcal{P}_{r,d}$	$\mathcal{P}_{2r,2d}$	$\mathcal{P}_{2r,2d}$	$\mathcal{P}_{4r,4d}$	$\mathcal{P}_{3r,3d}$
simulator	$\mathcal{P}_{r,d}$	$\mathcal{P}_{2r,2d}$	$\mathcal{P}_{r,d}$	$\mathcal{P}_{4r,4d}$	$\mathcal{P}_{3r,3d}$

Table 2. Protocol modification to accommodate simulation.

when the challenge is $b = 0$	S	V	R	X	Y
legitimate protocol	$\mathcal{P}_{r,d}$	$\mathcal{P}_{2r,2d}$	$\mathcal{P}_{2r,2d}$	$\mathcal{P}_{4r,4d}$	$\mathcal{P}_{2r,2d}$
simulator	$\mathcal{P}_{r,d}$	$\mathcal{P}_{2r,2d}$	$\mathcal{P}_{2r,2d}$	$\mathcal{P}_{4r,4d}$	$\mathcal{P}_{2r,2d}$

While this fixes the size problem, nothing guarantees that the (X, Y) distributions of the parties and of the simulator are strictly identical. Nonetheless this gives hope to prove the protocol secure in the statistical zero-knowledge rather than in the perfect zero-knowledge sense. We did not explore further this point.

Note that while CGS security relied on the conjectured hardness of factoring polynomials in $S[t]$, the Fiat-Shamir variant relies on both the hardness of factoring polynomials in $S[t]$ and on the conjectured hardness of computing square roots in $S[t]$, a problem which might turn out to be easier than factoring polynomials in $S[t]$ and which hardness we did not assess[3].

[3] Note, for instance that $2P(0) = (P \otimes P)(0)$ and that the same occurs on the leading coefficients of both P and $P \otimes P$. It is unclear if this can be used to unravel P from $P \otimes P$.

3 Fiat-Shamirization of Chen, Grigoriev and Shpilrain

We denote $\text{hash}(m, i) = H_i \in \mathcal{P}_{r,d}$.

Key generation remains unchanged with respect to the original CGS.

To sign a message, generate $\vec{U}, \vec{V} \in_R (\mathcal{P}_{r,d})^\tau$ and compute $(\vec{A}, \vec{B}, \vec{N})$ as in the original CGS. Let $\vec{C} = \vec{A} \otimes \vec{B}$. Here $\vec{C}$ will act as a non-interactive commitment on $\vec{A}$ and $\vec{B}$.

$$h = \text{hash}(r, d, m, M, \vec{C}, \vec{N}) \bmod 2^\tau$$

We start by including in the signature $\vec{N}$.

We now use the τ bits of h as indicators pointing which commitments to open.

$$\begin{cases} \text{if } \ h_i = 0 \text{ add to the signature } \ U_i, V_i, C_i \\ \text{if } \ h_i = 1 \text{ add to the signature } \ A_i, B_i \end{cases}$$

The verifier can hence reconstruct $\vec{C}$ in full. Either they get A_i, B_i and can hence compute C_i (case $h_i = 1$) or they get C_i directly (case $h_i = 0$). The verifier can hence recompute h from the signature.

At each $h_i = 1$ coordinate the verifier performs tests V1,V2,V3,V4 and V5.

At each $h_i = 0$ coordinate the verifier checks that $U_i, V_i \in \mathcal{P}_{r,d}$ and $N_i \in \mathcal{P}_{2r,2d}$.

This idea can come in several flavors e.g. a different X_i, Y_i can be used per coordinate. In a more daring variant we can aggregate the different signature components.

In this variant we modify the definition of h to:

$$h = \text{hash}(r, d, m, M, \bar{C}, \bar{N}) \bmod 2^\tau \ \text{ where } \ \bar{C} = \bigotimes_{i=0}^{\tau} C_i \ \text{ and } \ \bar{N} = \bigotimes_{i=0}^{\tau} N_i$$

Let:

$$\bar{A}^1 = \bigotimes_{h_i=1} A_i \ \text{ and } \ \bar{B}^1 = \bigotimes_{h_i=1} B_i \ \text{ and } \ \bar{C}^0 = \bigotimes_{h_i=0} (A_i \otimes B_i) = \bigotimes_{h_i=0} C_i$$

$$\bar{N}^0 = \bigotimes_{h_i=0} N_i \ \text{ and } \ \bar{N}^1 = \bigotimes_{h_i=1} N_i \ \text{ and } \ M^\tau = \bigotimes_{i=0}^{\tau-1} M$$

If the signature was correctly generated we should have:

$$\bar{A}^1 \otimes \bar{B}^1 \otimes \bar{C}^0 = \bar{N} \otimes M^\tau \otimes \bigotimes_{i=0}^{\tau} (H_i \otimes H_i)$$

$$\bar{A}^1 \otimes \bar{B}^1 \otimes \bar{C}^0 = \bar{N}^0 \otimes \bar{N}^1 \otimes M^\tau \otimes \bigotimes_{i=0}^{\tau} (H_i \otimes H_i)$$

We can hence provide as a signature:

$$\bar{A}^1, \bar{B}^1, \bar{C}^0, \bar{N}^1, \text{ plus all the } U_i, V_i \text{ couples for which } h_i = 0$$

How to generate U_i, V_i? In all the above we assumed for the sake of clarity that $\vec{U}, \vec{V}$ are randomly generated.

The second (aggregated) variant uses $\otimes$ as a hash function which is bad because $\otimes$ is commutative. We hence enforce an extra protection to thwart element permutation attacks (see e.g. [1]). The protection consists in generating each U_i, V_i pair from a common random seed σ_i and by including in the process transforming σ_i into U_i, V_i both σ_i **and the index** i. To reveal a given U_i, V_i pair the signer reveals σ_i. This protection is mandatory in the aggregated scheme and recommended as an extra precaution for the non-aggregated version.

4 Tropicalizing other cryptosystems

The "tropicalization" strategy described for Fiat-Shamir case *may* apply to a variety of classical cryptosystems as long as during all computations[4] the following holds:

- There is no need to divide and;
- Multiplication depth remains reasonable.

The first condition stems from the (conjectured) absence of efficient tropical division. The second is due to the fact that the degree and the coefficients of the involved polynomials grows as we keep $\otimes$ing.

At a first glance those conditions do not seem to apply to schemes such as Diffie-Hellman. Fortunately, two interesting observations may still still salvage the situation.

First we observe that if $P_i \in \mathcal{P}_{r,d}$ then:

$$\bigotimes_{i=0}^{\ell-1} P_i \in \mathcal{P}_{\ell r, \ell d}$$

It follows that even if multiplication depth is huge, e.g. in a tropical Diffie-Helman with 1024-bit exponents, coefficients will be large but manageable. The degree of the resulting polynomial is however problematic as, in the example given, we would end-up with polynomials of degree $(\ell d)^2 = 2^{2048} d^2$.

The workaround may consist in reducing the resulting polynomials modulo x^q, i.e. working in $S[t]/(t^q)$ chopping all terms whose degree exceeds q. e.g., one could consider $q = 10d$. Working modulo polynomials more complex than x^q (e.g. $S[t]/(t^q \pm 1)$) is yet another option and has the advantage of recycling the "most significant" information of the polynomials while preserving size.

[4] Be it signature, verification, encryption or decryption.

Such approaches would allow tropicalizing cryptosystems with high multiplication depth such as Diffie-Hellman. This rough and general blueprint requires a deeper analysis because the coefficients of the polynomial G^x in $S[x]$ are, in essence, very close to a small constant times x. Reducing each coefficient modulo some small prime modulus e seems to avoid this problem but might create others.

The case of El-Gamal variants where division is not required is interesting. Such variants exist (e.g. EG I.3 or EG I.4 in [4]) but now the s part of the signature must be given in $\mathbb{Z}$ which might be vulnerable and deserves further investigations and/or new countermeasures.

5 Acknowledgments

We thank Dan Brown and Chris Monico on their great insight and pertinent comments during the developments of the ideas listed in this paper.

References

1. F. Benhamouda, H. Ferradi, R. Géraud, and D. Naccache. Non-interactive provably secure attestations for arbitrary rsa prime generation algorithms. Cryptology ePrint Archive, Paper 2017/640, 2017. https://eprint.iacr.org/2017/640.
2. J. Chen, D. Grigoriev, and V. Shpilrain. Tropical cryptography iii: digital signatures. Cryptology ePrint Archive, Paper 2023/1475, 2023. https://eprint.iacr.org/2023/1475.
3. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Proceedings on Advances in Cryptology—CRYPTO '86*, page 186–194, Berlin, Heidelberg, 1987. Springer-Verlag.
4. P. Horster, H. Petersen, and M. Michels. Meta-elgamal signature schemes. In *Proceedings of the 2nd ACM Conference on Computer and Communications Security*, CCS '94, page 96–107, New York, NY, USA, 1994. Association for Computing Machinery.
5. K. H. Kim and F. W. Roush. Factorization of polynomials in one variable over the tropical semiring, 2005.
6. A. Muanalifah and S. Sergeev. On the tropical discrete logarithm problem and security of a protocol based on tropical semidirect product. *Communications in Algebra*, 50(2):861–879, Sept. 2021.
7. L. Panny. Forging tropical signatures. Cryptology ePrint Archive, Paper 2023/1748, 2023. https://eprint.iacr.org/2023/1748.