More forging (and patching) of tropical signatures

Daniel R. L. Brown* and Chris Monico†

November 29, 2023

Abstract

Panny [3] described how to forge the "tropical signatures" proposed by Chen, Grigoriev and Shpilrain [1]. (These signatures are loosely related to the NP-complete problem of factoring tropical polynomials).

We describe more methods to forge these tropical signatures. We also describe some patches that thwart all but one of these forgery methods (which we summarize as re-hashing an honest signature).

1 Introduction

In [1], a cryptographic signature scheme is proposed using polynomials over a tropical semiring.

We review the proposed signature scheme. Next, we describe several methods to forge these signatures. After this, we propose some patches that thwart all but one of these forgery methods.

2 Tropical signatures: review

Tropical signatures were proposed in [1]. We review this proposal, in two steps, and then discuss a generalization (which also is affected by some of our forgeries).

2.1 Tropical polynomials

Tropical signatures use tropical polynomials, which we review briefly.

Consider the semiring $(S, \oplus, \otimes) = (\mathbb{Z} \cup \{\infty\}, \min, +)$. Since integer multiplication plays no role in S, for $s \in S$ and $n \in \mathbb{N}$, we denote

$$s^n = \underbrace{s \otimes s \otimes \ldots \otimes s}_{n \text{ copies}} = \underbrace{s + s + \cdots + s}_{n \text{ copies}} = ns.$$

*BlackBerry, danibrown@blackberry.com

†Texas Tech, c.monico@ttu.edu

Some authors have preferred the notation $s^{\otimes n}$ or $s^{(n)}$, but since no confusion will arise here we will use the simpler notation s^n as above.

The *polynomial semiring* $S[t]$ is the set of formal sums

$$S[t] = \left\{ \bigoplus_{j=0}^{n} s_j \otimes t^j \ : \ n \geq 0, s_0, \ldots, s_n \in S \right\}.$$

It can be shown that $S[t]$ is also a commutative semiring.

For a polynomial $f(t) = f_0 \ \oplus \ f_1 \otimes t \ \oplus \ f_2 \otimes t^2 \ \oplus \ \ldots \ \oplus \ f_m \otimes t^m$, we define $\deg f$ to be the greatest integer $n \leq m$ for which $f_n \neq \infty$, if such an n exists.

2.2 A signature scheme

The signature scheme proposed in [1] is as follows.

1. Alice chooses polynomials $X, Y \in S[t]$, each with coefficients in $[0, r]$ and degree d. She computes $M = X \otimes Y$ and publishes her *public key* (r, d, M).

2. To sign a message, Alice uses a standard hash function to produce a corresponding hash[1] $H \in S[t]$, with coefficients in $[0, r]$ and degree d.

3. Alice chooses two random polynomials $U, V \in S[t]$ with coefficients in $[0, r]$ and degree d. Set $N = U \otimes V$. The signature for this message is the 4-tuple

$$\Big(H, \ H \otimes X \otimes U, \ H \otimes Y \otimes V, \ N \Big).$$

One verifies Alice's signature (H, A, B, N) as follows.

(V1) Verify that H has degree d and coefficients in $[0, r]$, and that H is indeed the correct hash of the message.

(V2) Verify that A and B each have degree $3d$ and coefficients in $[0, 3r]$.

(V3) Verify that N has degree $2d$ and coefficients in $[0, 2r]$.

(V4) Verify that neither B nor C is a constant tropical multiple of $H \otimes M$ or $H \otimes N$.

(V5) Verify that $A \otimes B = H \otimes H \otimes M \otimes N$.

2.3 Generalization of the signatures

One of our forgery methods is general enough to apply to a generalization of tropical signatures. The generalized signature is described below.

Replace the set of tropical polynomials by some other set. Replace multiplication of tropical polynomial by some other operation, also written as $\otimes$. Use a similar signing and verification process.

[1] In [1], variable P is used for the hash, but we opt to use H.

If $\otimes$ is commutative and associative[2], then condition (V5) still holds:

$$\begin{aligned} A \otimes B &= (H \otimes X \otimes U) \otimes (H \otimes Y \otimes V) \\ &= (H \otimes H) \otimes (X \otimes Y) \otimes (U \otimes V) \\ &= H \otimes H \otimes M \otimes N. \end{aligned}$$

Further details are needed to fully specify an instance of this generalization:

- How the signer samples X, Y, U and V (secretly).
- How the signer and verifier compute H as a hash of a message.
- How the verifier does membership tests (V1), (V2), and (V3).
- How the verifier does the divisibility test (V4).

Security depends on all these details. Cryptographers are safer to presume that most practical instances of this generalization are insecure, until proven otherwise.

If a previously published signature scheme was an instance of this generalization, then the generalization should be named after this. If not, then the generalization might merit a name, with credit to [1]. Because the signer generates a factored nonce $N = U \otimes V$, it seems reasonable to call this generalization a *factored nonce* signature.

3 Tropical polynomial division

In this section, we present some definitions, notations, and results which will be used in later sections.

We define an order $\preccurlyeq$ on S by $x \preccurlyeq y$ if $x \oplus y = x$. This extends to a partial order on $S[t]$ by $a \preccurlyeq b$ if $\deg a = \deg b$ and $a \oplus b = a$. Note that $a \preccurlyeq b$ if and only if $\deg a = \deg b$ and every coefficient of a is less or equal the corresponding coefficient of b.

This partial order respects both operations of $S[t]$: if $a \preccurlyeq b$ and $c \in S[t]$, then

$$\Big(a \oplus c\Big) \oplus \Big(b \oplus c\Big) = a \oplus c,$$

so $a \oplus c \preccurlyeq b \oplus c$. Similarly, if $a \preccurlyeq b$ then $a \otimes c \preccurlyeq b \otimes c$.

For polynomials $a, b \in S[t]$, we define

$$b \,\big|\, a, \qquad \text{if } a = b \otimes q, \qquad \text{for some } q \in S[t].$$

The following is described in [2, Section 6], but we provide a complete self-contained argument here.

Proposition 3.1 (Exact division of polynomials)**.** *Suppose that $a, b \in S[t]$. Let $m = \deg(b)$ and $n = \deg(a) - m$. For $0 \le j \le n$ let*

$$q_j = \max_{0 \le i \le m} \{a_{i+j} - b_i\},$$

and $q(t) = q_0 \ \oplus\ q_1 \otimes t \ \oplus\ \ldots\ \oplus\ q_n \otimes t^n$. Then

[2] More generally, it suffices that $\otimes$ is medial, instead of being commutative and associative.

1. $a \preccurlyeq b \otimes q$*.*

2. If $d \in S[t]$ *with degree* n *and* $a \preccurlyeq b \otimes d$*, then* $b \otimes q \preccurlyeq b \otimes d$*.*

3. If $b \,\big|\, a$*, then* $a = b \otimes q$*.*

Proof. Let a, b, q, m, n be as above, and let $p = b \otimes q$. Then for $0 \leq k \leq m+n$ we have that $p_k = \min\limits_{i+j=k} \{b_i + q_j\}$, Let $i_0 \in \{0, 1, \ldots, m\}$ and $j_0 \in \{0, 1, \ldots, n\}$. Then

$$a_{i_0+j_0} - b_{i_0} \leq \max_{0 \leq i \leq m} \{a_{i+j_0} - b_i\} = q_{j_0},$$

and so $a_{i_0+j_0} \leq b_{i_0} + q_{j_0}$. In particular, for each $0 \leq k \leq m+n$, this holds for all i_0, j_0 with $i_0 + j_0 = k$, which implies that $a_k \leq p_k$ for all such k. Therefore $a \preccurlyeq p$.

We prove Part 2 by contrapositive. Suppose $d \in S[t]$ has degree n and $b \otimes q \not\preccurlyeq b \otimes d$. If $q \preccurlyeq d$ we would have $b \otimes q \preccurlyeq b \otimes d$, so we must have $q \not\preccurlyeq d$. It follows that $q_{j_0} > d_{j_0}$ for at least one $j_0 \in \{0, 1, \ldots, n\}$. By definition of q_{j_0} there exists $i_0 \in \{0, 1, \ldots, m\}$ for which

$$d_{j_0} < q_{j_0} = a_{i_0+j_0} - b_{i_0}.$$

Letting $k_0 = i_0 + j_0$ we have that $a_{k_0} > b_{i_0} + d_{j_0} \geq \min\limits_{i+j=k_0} \{b_i + d_j\}$, so that $a \not\preccurlyeq b \otimes d$.

Now suppose that $b \,\big|\, a$. Then there exists $d \in S[t]$ with $\deg(d) = n$ [3] such that $a = b \otimes d$. By Parts 1 and 2 we have that $a \preccurlyeq b \otimes q \preccurlyeq b \otimes d = a$, and hence $b \otimes q = a$. □

The following result is obvious, but nevertheless worth making explicit.

Proposition 3.2. *Let* $f, g \in S[t]$*. If the coefficients of* f *lie in* $[0, a]$ *and the coefficients of* g *lie in* $[0, b]$*, then the coefficients of* $f \otimes g$ *lie in* $[0, a+b]$*.*

4 Forgery methods

This section presents several methods to forge tropical signatures, including the forgery from [1] that is thwarted by step (V4) from [1].

4.1 Swapping symbols Y and U

The following attack was already presented in [1], and was already thwarted by countermeasure (V4) from [1]. We review this attack, because we found a very similar attack, not thwarted by any steps from [1].

In short, this forger does the exact same steps as the signer, except that the forger swaps the symbols Y and U in the computation of the signature:

$$(H,\ H \otimes X \otimes Y,\ H \otimes U \otimes V,\ U \otimes V).$$

[3] A bit of care and detail in the definitions is required to get this, but if we define polynomials and degree correctly, this should be true.

The forger does not know the values of symbols X or Y, but the forger does know $X \otimes Y = M$. Because symbols X and Y have now been put together, the forger can compute the actual values of the formula above as:

$$(H,\ H \otimes M,\ H \otimes U \otimes V,\ U \otimes V).$$

Verification checks (V1), (V2) and (V3) will hold, because the four variables X, Y, U and V are sampled from the same subset. Verification check (V5) will hold, because $\otimes$ is commutative and associative.

Verification check (V4) will stop this forgery. Indeed, the purpose of (V4) is to thwart this forgery. Notably:

- This forger is a universal forger. The forger can sign any message you want it to.
- This forger is a key-only forger, also known as no-message. The forger can produce a forgery given only the public-key. The forger does not need the signer's help by way of generating honest signatures for the forger to exploit.
- This forgery should work against any suitable generalization of tropical signatures, provided it has $\otimes$ commutative and associative, and it lacks a suitable generalization of verification check (V4). In other words, this forgery likely breaks most forms of factored nonce signatures.
- The forger does not need to divide any tropical polynomials.
- The countermeasure (V4) to this attack requires the verifier to implement some form of divisibility test, which is specific to the choice of $\otimes$. By contrast to the forger does not need to divide. This makes the forger more generic and generally faster than the signer.

4.2 Swapping symbols Y and H

A forger similar to the previous works by swapping symbols Y and H. The resulting forgery is:

$$(H,\ M \otimes U,\ H \otimes H \otimes V,\ U \otimes V),$$

where, once again, the symbol Y has been put together with symbol X, to give the public key M, which the forger knows.

This forger is not prevented by any of the verification steps (V1), ..., (V5) from [1]. Like the previous method, the forger does not need to divide, and the forger can be applied to any generalization (a factored nonce signature) with $\otimes$ commutative and associative.

Though this forger is quite similar to the previous one, it seems to have been overlooked in [1].

4.3 Factoring the hash

We present here a method which can be used to forge signatures for around 2% of all possible hash values $H(t)$, with the parameter sizes $d = 150$ and $r = 127$. This success rate is already non-negligible, but can sometimes become even more serious. In many real world settings, if one wishes to forge a signature for a specific message, it may be easy to produce a few hundred message variants conveying the same essential information but having different hash values. One of the variant messages is likely to have a hash susceptible to a forgery.

Suppose that (r, d, M) is a public key of the form specified in the introduction. Let $H \in S[t]$ with $\deg H = d$, coefficients in $[0, r]$, and having a degree 1 divisor L whose coefficients are in $[0, r]$. Then $H = L \otimes Q$ for some Q which may be found using Proposition 3.1. Let ℓ be the maximum finite coefficient of L and let q be the maximum finite coefficient of Q. Let $U \in S[t]$ with $\deg U = d - 1$ and coefficients in $[0, r - \ell]$. Let $V \in S[t]$ with $\deg V = d + 1$ and coefficients in $[0, r - q]$. Then with $N = U \otimes V$, we find that

$$(H,\ L \otimes M \otimes U,\ Q \otimes H \otimes V,\ N),$$

is a valid signature. Certainly the degrees are correct and the product condition is met. Moreover, it is highly likely that neither $L \otimes M \otimes U$ nor $Q \otimes H \otimes V$ is a constant (tropical) multiple of $H \otimes M$ or $H \otimes N$. The coefficients of each component are all in the proper range, by Proposition 3.2.

For parameter sizes in the range of interest, $d = 150$ and $r = 127$, experimental evidence suggests that around 2% of all hash polynomials H have a linear divisor L with integer coefficients in the interval $[0, r]$. Of course, one may also consider many such H polynomials which are constructed with such a divisor.

4.4 Factoring the public key

If an attacker can find a proper divisor D of M with coefficients in $[0, r]$, then she can forge a signature as follows. She uses Proposition 3.1 to find Q such that $M = D \otimes Q$, and chooses a hash $H \in S[t]$ with $\deg H = d$ and coefficients in $[0, r]$. Let q be the maximum finite coefficient of Q. She chooses $U \in S[t]$ with coefficients in $[0, r]$ and $\deg U = \deg Q$. She also chooses $V \in S[t]$ with coefficients in $[0, r - q]$ and $\deg V = \deg D$. Then the 4-tuple

$$(H,\ H \otimes D \otimes U,\ H \otimes Q \otimes V,\ U \otimes V),$$

is a valid signature. A decomposition $M = D \otimes Q$ is effectively usable by an eavesdropper in a way similar to how the legitimate signer uses $M = X \otimes Y$.

To find such a proper divisor D of M, one may simply proceed by brute force. The lack of unique factorization in $S[t]$ is pronounced, and this is non-negligible. For 300 randomly chosen pairs $X, Y \in S[t]$ with coefficients in $[0, 127]$ and $\deg X = \deg Y = 150$, we found that $X \otimes Y$ had a degree 1 divisor with coefficients in $[0, r]$ in 143 of these 300 cases.

The main point here is that it is not necessary for an attacker to obtain the particular factorization $M = X \otimes Y$; there are generally many possible non-trivial factorizations, many of which will allow her to forge a signature.

4.5 Panny's forgeries

While this paper was under preparation, Lorenz Panny [3] independently proposed several method for forging tropical signatures.

4.5.1 Tweaking the symbol swap forgery

Panny's first forgery method can be considered a tweak of the forgery that swaps Y and U. Recall that this symbol swap forgery computes $A = H \otimes M$, $B = H \otimes N$. But this forgery cannot be directly used as-is to form a signature because the protocol checks at (V4) whether this is the case. However, Panny observed that the non-cancellative multiplication $\otimes$ makes it fairly easy to slightly vary the coefficients of A and B, obtaining A', B' which are not constant tropical multiples of $H \otimes M$ and $H \otimes N$ respectively, yet still satisfy $A' \otimes B' = H^2 \otimes M \otimes N$.

4.5.2 Double dividing

Indeed, an additional way this could be done is to set A' to be the quotient of $H^2 \otimes M \otimes N$ by B, as given by Proposition 3.1; similarly set B' to be the quotient of $H^2 \otimes M \otimes N$ by A'.

4.5.3 Factoring $H \otimes M$

Panny's second forgery method is essentially to factor $H \otimes M$. This works similarly to our attacks that factor H or that factor M. Search for a hash polynomial H for which $H \otimes M$ has a proper divisor that can be quickly found. In practice, this is not hard to do - it is, in fact, easier than what we suggested in the previous sections. Assuming this to be the case, apply Proposition 3.1 to write $H \otimes M = D_1 \otimes R_1$, with $\deg D_1 < \deg R_1$. Construct a nonce N for which $H \otimes N = D_2 \otimes R_2$ for some $D_2, R_2 \in S[t]$ with $\deg D_2 = \deg D_1$, and with D_2, R_2 having coefficients in appropriate ranges. It follows that $(H, D_1 \otimes R_2, D_2 \otimes R_1, N)$ is a valid signature.

4.6 Re-hashing an honest signature

Suppose that an attacker knows the public key (r, d, M) and a valid signature

$$\Big(H,\ H{\otimes}X{\otimes}U,\ H{\otimes}Y{\otimes}V,\ N\Big).$$

with $H, X, Y, U, V, N = U \otimes V$ as in Section 1. Given the division technique described in Section 3, if the divisions of A and B by H yielded $X \otimes U$ and $Y \otimes V$ respectively, it would be easy to forge a signature for an arbitrary hash value $\widetilde{H}$. Experiments suggest that this exact division occurs at both A and B approximately 0.5% of the time. One can therefore expect that, after seeing approximately 200 signatures, the forger can then forge any message.

Typically, given a single signature only, the quotients obtained are strictly less than $X \otimes U$ and $Y \otimes V$ respectively. So, approximately 99.5% of the time, given only a single signature, this basic forgery method fails. Nevertheless, the basic method above is the starting point for a more sophisticated single-signature attack.

Suppose the attacker can find $Q_A, Q_B \in S[t]$ each with degree $2d$, with coefficients in $[0, 2r]$, satisfying $Q_A \otimes Q_B = M \otimes N$, and neither Q_A nor Q_B being a constant tropical multiple of M or N; such a pair certainly exists since $Q_A = X \otimes U$ and $Q_B = Y \otimes V$ is one such pair. Then for an arbitrary $\widetilde{H} \in S[t]$ with degree d and coefficients in $[0, r]$, it's not hard to see that the 4-tuple

$$\left(\widetilde{H},\ \widetilde{H} \otimes Q_A,\ \widetilde{H} \otimes Q_B,\ N\right)$$

is also a valid signature for the public key (r, d, M).

An attacker may attempt find such polynomials Q_A, Q_B as follows. For $f \in S[t]$ and $j \leq \deg f$, we let $(f)_j$ denote the coefficient of t^j. Set $C = M \otimes N$, and set q_A, q_B to be the minimal quotients of A and B by H, as produced by Proposition 3.1. Then we have that $H \otimes q_A = A$, $H \otimes q_B = B$, and $q_A \preccurlyeq X \otimes U$ and $q_B \preccurlyeq Y \otimes V$. Therefore $q_A \otimes q_B \preccurlyeq C$. If we had $q_A = X \otimes U$ and $q_B = Y \otimes V$, we would be done. However, this occurs quite rarely for parameter values in the range of interest. The general situation that arises is $q_A \prec X \otimes U$ and $q_B \prec Y \otimes V$. The idea is simply to increase coefficients of q_A and/or q_B, as necessary, until $q_A \otimes q_B = C$.

For each $i, j \in \{0, 1, \ldots, 2d\}$, set $\delta_{ij} = C_{i+j} - (q_A)_i - (q_B)_j$. Repeat the following, while $q_A \otimes q_B \prec C$. Let k be the least non-negative integer for which $q_A \otimes q_B$ and C disagree in coefficient k. For each $i + j = k$, if $\delta_{ij} > 0$ then first try increasing $(q_A)_i$ by δ_{ij}. Doing so, if we still have $q_A \otimes q_B \preccurlyeq C$, then keep this change of q_A. Otherwise, leave q_A unchanged and try increasing $(q_B)_j$ by δ_{ij}. If we still have $q_A \otimes q_B \preccurlyeq C$, then keep this change of q_B; if not, report a failure.

We randomly generated 10000 public keys with the parameters $d = 150$ and $r = 127$ suggested in [1]. For each public key, we generated one valid signature randomly and applied the technique above to create a forged signature for a randomly chosen hash. In these 10000 attempts, 9965 resulted in successful forgeries which passed the verification tests (V1)–(V5).

5 Patches to tropical signatures

This section describes some patches to the original tropical signatures. Each patch was devised narrowly to thwart one of our forgery attacks. So, the patches are not part of an effort to save tropical signatures. Rather, the patches save us from overstating the reach of our forgery results. We have not yet found a patch against our re-hashing the honest signature. Neither our attacks nor our patches should be expected withstand further patches or attacks (respectively).

5.1 A new verification check of divisibility

One patch is to apply another verification test, in addition to those from [1], which were enumerated in Section 2.2.

(V6) Verify that H divides both A and B, and that M divides neither A nor B, and that N divides neither A nor B.

This condition may be verified using Proposition 3.1. This patch has the following impacts on the previously discussed forgery methods.

- *Swapping symbols Y and U*, from Section 4.1: Those forgeries have component A divisible by M, and are therefore thwarted by (V6).
- *Swapping symbols Y and H*, from Section 4.2: Those forgeries also have component A divisible by M, and are therefore thwarted by (V6).
- *Factoring the hash*, from Section 4.3: Those forgeries have component A which is divisible by M; also, component A is typically not divisible by H. Thus, these are also thwarted by (V6).
- *Tweaking the symbol swap*, from Section 4.5.1: Varying the coefficients of $H \otimes M$ typically yields a result which is not divisible by H. These are therefore thwarted by (V6).
- *Double dividing*, from Section 4.5.2: This is just a very special case of the previous, and so it fails to pass (V6) for the same reason(s).

5.2 Leading and trailing zeros

A second patch is to choose the private key values X and Y such that the first and last coefficients of X and Y are zero. We have that the first and last coefficients of $M = X \otimes Y$ are also zero. It's easy to see in that case that if D is any divisor of M with coefficients in $[0, r]$, then D must also have first and last coefficients of zero. Thus, the only possible degree 1 divisor would be $D(t) = 0 \ \oplus \ 0 \otimes t$, and it seems (empirically) that this will not divide M in the generic case.

This patch seems to thwart, on its own, the following attacks.

- *Factoring the public key*, from Section 4.4: By construction, H will divide the second and third signature components, and it is highly unlikely that either are divisible by M or N. So (V6) alone does not defeat this forgery technique. However, if X and Y have first and last coefficient of zero, it seems to be difficult to find a divisor of $M = X \otimes Y$; at least, a brute force approach enumerating possible divisors of small degree with first and last coefficients zero seems to not generally find such a divisor.

This patch when combined with (V6) patch seems to thwart Panny's attacks.

- *Factoring $H \otimes M$*, from Section 4.5.3: We used the code made available by [3], adding condition (V6) to the verification code. For 1000 randomly generated keys, 358 of the forged signatures still passed. So the additional verification test (V6) alone is insufficient to defend against Panny's attack. However, when we also imposed the requirement that the first and last coefficients of both X and Y be zero, and repeated the experiment, none of the 1000 forged signatures were accepted as valid. Specifically, all of the resulting forgeries failed to have second and third components divisible by H. It is entirely possible, though, that this technique might be modified to still work in the present situation.

5.3 Re-hashing honest signature survives

The technique presented in Section 4.6 survives, even with the additional verification test (V6) and X, Y chosen with first and last coefficients of zero. By construction, the second and third components of the forged signature are each divisible by the first component. Nothing in the construction seems to make it particularly likely that the second or third components will be divisible by M or N. Finally, the construction we suggest for obtaining Q_A and Q_B seems to be unaffected when X, Y are chosen to have first and last coefficients of zero.

We randomly generated 10000 public keys, from X and Y having first and last coefficients of zero, with the suggested parameter values [1] of $d = 150$ and $r = 127$. For each public key, we generated one valid signature. For each of these, we choose a random hash value and employed the technique in Section 4.6 to create a forged signature. Of these 10000 experiments, we were able to produce 9963 forged signatures which passed the verification tests, including the additional verification test suggested in Section 5.1. This forgery technique therefore appears to be almost completely unaffected by the extra verification test (V6) and the restricted choices of X and Y.

6 Conclusion

We found several attacks on tropical signatures. We also presented some straightforward countermeasures for preventing some of these - namely, the extra verification condition (V6) and restricting the private polynomials X, Y to have first and last coefficients zero. While these countermeasures also apparently thwart the techniques from [3], they do not prevent our last forgery technique from Section 4.6. Indeed, we have not found any way to prevent that type of forgery.

References

[1] Jiale Chen, Dima Grigoriev, and Vladimir Shpilrain. Tropical cryptography III: digital signatures. Cryptology ePrint Archive, Paper 2023/1475, 2023. https://eprint.iacr.org/2023/1475.

[2] Ki Hang Kim and Fred W. Roush. Factorization of polynomials in one variable over the tropical semiring, 2005. https://arxiv.org/abs/math/0501167.

[3] Lorenz Panny. Forging tropical signatures. Cryptology ePrint Archive, Paper 2023/1748, 2023. https://eprint.iacr.org/2023/1748.