Forging tropical signatures

Lorenz Panny

Technische Universität München

AAC'24, Abu Dhabi, 8 March 2024

TROPICAL CRYPTOGRAPHY III: DIGITAL SIGNATURES

JIALE CHEN, DIMA GRIGORIEV, AND VLADIMIR SHPILRAIN

Abstract. We use tropical algebras as platforms for a very efficient digital signature protocol. Security relies on computational hardness of factoring one-variable tropical polynomials; this problem is known to be NP-hard.

This talk

How to break tropical signatures (in several different ways)

Some comments on cryptographic design methodology

This talk

How to break tropical signatures (in several different ways)

Some comments on cryptographic design methodology

Blueprint of the construction

Let S be a commutative semigroup: $(ab)c = a(bc)$ and $ab = ba$.

Blueprint of the construction

Let S be a commutative semigroup: $(ab)c = a(bc)$ and $ab = ba$.

Candidate signature scheme(**?**):

- Private key: $x, y \xleftarrow{\text{random}} S$.
- Public key: $m = xy \in S$.
- Signing: Let $h \in S$ be a message hash.
  Pick $u, v \xleftarrow{\text{random}} S$, return $(s_1, s_2, n) := (hxu, hyv, uv)$
- Verifying: Check $s_1 s_2 = hhmn$.

Blueprint of the construction

Let S be a commutative semigroup: $(ab)c = a(bc)$ and $ab = ba$.

Candidate signature scheme(**?**):

- Private key: $x, y \xleftarrow{\text{random}} S$.
- Public key: $m = xy \in S$.
- Signing: Let $h \in S$ be a message hash.
  Pick $u, v \xleftarrow{\text{random}} S$, return $(s_1, s_2, n) := (hxu, hyv, uv)$
- Verifying: Check $s_1 s_2 = hhmn$.

Idea: Key recovery means recovering (x, y).

- Path A: Factor m into x, y.
- Path B: Factor n into u, v; find x, y from hxu, hu and hyv, hv.

Blueprint of the construction

Let S be a commutative semigroup: $(ab)c = a(bc)$ and $ab = ba$.

Candidate signature scheme(**?**):

- Private key: $x, y \xleftarrow{\text{random}} S$.
- Public key: $m = xy \in S$.
- Signing: Let $h \in S$ be a message hash.
  Pick $u, v \xleftarrow{\text{random}} S$, return $(s_1, s_2, n) := (hxu, hyv, uv)$
- Verifying: Check $s_1 s_2 = hhmn$.

Idea: Key recovery means recovering (x, y).

- Path A: Factor m into x, y.
- Path B: Factor n into u, v; find x, y from hxu, hu and hyv, hv.

Q: What about forgery attacks that do not recover (x, y)?

$\rightsquigarrow$ Significantly more ad-hoc problem.

Tropical algebra

Core object: The tropical semiring.

Tropical algebra

Core object: The tropical semiring.

It consists of the set $\mathbb{T} := \mathbb{R} \cup \{\infty\}$ with two binary operations:

- "$\oplus$", which is ordinary $\min$.
- "$\otimes$", which is ordinary $+$.

Tropical algebra

Core object: The tropical semiring.

It consists of the set $\mathbb{T} := \mathbb{R} \cup \{\infty\}$ with two binary operations:

- "$\oplus$", which is ordinary $\min$.
- "$\otimes$", which is ordinary $+$.

Some properties:

- $(\mathbb{T}, \oplus)$ is a commutative monoid with neutral element ∞.
- $(\mathbb{T}, \otimes)$ is a commutative monoid with neutral element 0.
- The distributive law holds: $a \otimes (b \oplus c) = a \otimes b \oplus a \otimes c$.
- Absorption properties: $a \oplus a = a$ and $\infty \otimes a = \infty$.

Tropical polynomials

Consider symbolic polynomials over $\mathbb{T}$:

$$F(x) = c_0 \oplus (c_1 \otimes x) \oplus (c_2 \otimes x \otimes x) \oplus \cdots \oplus (c_n \otimes x^{\otimes n}) .$$

with all $c_i \in \mathbb{T}$.

Tropical polynomials

Consider symbolic polynomials over $\mathbb{T}$:

$$F(x) = c_0 \oplus (c_1 \otimes x) \oplus (c_2 \otimes x \otimes x) \oplus \cdots \oplus (c_n \otimes x^{\otimes n})\,.$$

with all $c_i \in \mathbb{T}$. In more conventional notation:

$$F(x) = \min\{c_0, c_1 + x, c_2 + 2x, ..., c_n + nx\}\,.$$

(Note: “Missing” coefficients are ∞, not 0!)

Tropical polynomials

Consider symbolic polynomials over $\mathbb{T}$:

$$F(x) = c_0 \oplus (c_1 \otimes x) \oplus (c_2 \otimes x \otimes x) \oplus \cdots \oplus (c_n \otimes x^{\otimes n}) \,.$$

with all $c_i \in \mathbb{T}$. In more conventional notation:

$$F(x) = \min\{c_0, c_1 + x, c_2 + 2x, \ldots, c_n + nx\} \,.$$

(Note: "Missing" coefficients are ∞, not 0!)

Arithmetic works as usual, but with $(\oplus, \otimes)$ instead of $(+, \cdot)$.

- Example: $$\big(1 \oplus (3 \otimes x)\big) \otimes \big(-1 \oplus (2 \otimes x)\big)$$ $$= \; 0 \oplus (2 \otimes x) \oplus (5 \otimes x^{\otimes 2})$$

NP-hardness of tropical polynomial factorization

- Kim–Roush (2005, arXiv:math/0501167):
  Factoring tropical polynomials is **NP**-hard.
  Here "factoring" really means "splitting into a nontrivial product".

Proposed tropical signatures

Idea: As before, but now with multiplication of tropical polynomials, since factoring them is supposedly hard.

Proposed tropical signatures

Idea: As before, but now with multiplication of tropical polynomials, since factoring them is supposedly hard.

- Parameters: Two integers d, r. (Paper: $d = 150$ and $r = 127$.)
- Let $T_{d,r}$ denote the set of tropical polynomials of degree d with all coefficients in $\{0, ..., r\}$ and let $H\colon \{0,1\}^* \to T_{d,r}$.

Proposed tropical signatures

Idea: As before, but now with multiplication of tropical polynomials, since factoring them is supposedly hard.

- Parameters: Two integers d, r. (Paper: $d = 150$ and $r = 127$.)
- Let $T_{d,r}$ denote the set of tropical polynomials of degree d with all coefficients in $\{0, ..., r\}$ and let $H\colon \{0,1\}^* \to T_{d,r}$.
- Private key: Two tropical polynomials $X, Y \xleftarrow{\text{random}} T_{d,r}$.
- Public key: The tropical product $M := X \otimes Y$.

Proposed tropical signatures

Idea: As before, but now with multiplication of tropical polynomials, since factoring them is supposedly hard.

- Parameters: Two integers d, r. (Paper: $d = 150$ and $r = 127$.)
- Let $T_{d,r}$ denote the set of tropical polynomials of degree d with all coefficients in $\{0, ..., r\}$ and let $H\colon \{0,1\}^* \to T_{d,r}$.
- Private key: Two tropical polynomials $X, Y \xleftarrow{\text{random}} T_{d,r}$.
- Public key: The tropical product $M := X \otimes Y$.
- Signature: Three tropical polynomials S_1, S_2, N such that
  - $S_1, S_2 \in T_{3d,3r}$ and $N \in T_{2d,2r}$.
  - $S_1 \otimes S_2 = P \otimes P \otimes M \otimes N$ where $P = H(\mathit{message})$.
  - S_1, S_2 are not constant tropical multiples of $P \otimes M$ or $P \otimes N$.

Proposed tropical signatures

Idea: As before, but now with multiplication of tropical polynomials, since factoring them is supposedly hard.

- Parameters: Two integers d, r. (Paper: $d = 150$ and $r = 127$.)
- Let $T_{d,r}$ denote the set of tropical polynomials of degree d with all coefficients in $\{0, ..., r\}$ and let $H\colon \{0,1\}^* \to T_{d,r}$.
- Private key: Two tropical polynomials $X, Y \xleftarrow{\text{random}} T_{d,r}$.
- Public key: The tropical product $M := X \otimes Y$.
- Signature: Three tropical polynomials S_1, S_2, N such that
  - $S_1, S_2 \in T_{3d,3r}$ and $N \in T_{2d,2r}$.
  - $S_1 \otimes S_2 = P \otimes P \otimes M \otimes N$ where $P = H(message)$.
  - S_1, S_2 are not constant tropical multiples of $P \otimes M$ or $P \otimes N$.
- Honest signature: Sample $U, V \xleftarrow{\text{random}} T_{d,r}$ and let $N = U \otimes V$, $S_1 = P \otimes X \otimes U$, and $S_2 = P \otimes Y \otimes V$.

Warmup: "Trivial forgeries"

Recall: We require $S_1 \otimes S_2 = P \otimes P \otimes M \otimes N$,
such that $S_1, S_2 \in T_{3d,3r}$. (Recall $P \in T_{d,r}$ and $M, N \in T_{2d,2r}$.)

Warmup: "Trivial forgeries"

Recall: We require $S_1 \otimes S_2 = P \otimes P \otimes M \otimes N$,
such that $S_1, S_2 \in T_{3d,3r}$. (Recall $P \in T_{d,r}$ and $M, N \in T_{2d,2r}$.)

Easy: $S_1 = P \otimes M = P \otimes X \otimes Y$ and $S_2 = P \otimes N = P \otimes U \otimes V$.

Compare honest signature: $S_1 = P \otimes X \otimes U$ and $S_2 = P \otimes Y \otimes V$.

Warmup: "Trivial forgeries"

Recall: We require $S_1 \otimes S_2 = P \otimes P \otimes M \otimes N$,
such that $S_1, S_2 \in T_{3d,3r}$. (Recall $P \in T_{d,r}$ and $M, N \in T_{2d,2r}$.)

Easy: $S_1 = P \otimes M = P \otimes X \otimes Y$ and $S_2 = P \otimes N = P \otimes U \otimes V$.

Compare honest signature: $S_1 = P \otimes X \otimes U$ and $S_2 = P \otimes Y \otimes V$.

Also, can scale (S_1, S_2, N) by $(c_1, c_2, c_1 \otimes c_2)$ where $c_1, c_2 \in \mathbb{T}$.

Warmup: "Trivial forgeries"

Recall: We require $S_1 \otimes S_2 = P \otimes P \otimes M \otimes N$,
such that $S_1, S_2 \in T_{3d,3r}$. (Recall $P \in T_{d,r}$ and $M, N \in T_{2d,2r}$.)

Easy: $S_1 = P \otimes M = P \otimes X \otimes Y$ and $S_2 = P \otimes N = P \otimes U \otimes V$.

Compare honest signature: $S_1 = P \otimes X \otimes U$ and $S_2 = P \otimes Y \otimes V$.

Also, can scale (S_1, S_2, N) by $(c_1, c_2, c_1 \otimes c_2)$ where $c_1, c_2 \in \mathbb{T}$.

These "trivial forgeries" are why the verifier checks that S_1, S_2 aren't constant multiples of $P \otimes M$, $P \otimes N$.

Attack #1: Morphing products

- Observation:
  Tropical polynomial arithmetic is highly non-cancellable.

Attack #1: Morphing products

- Observation:
  Tropical polynomial arithmetic is highly non-cancellable.

- Example: Let $F(x) := \bigoplus_i c_i \otimes x^{\otimes i}$ and $G(x) = \bigoplus_i c'_i \otimes x^{\otimes i}$.
  Then the n^{th} coefficient d_k of $F(x) \otimes G(x)$ looks like
  $$\min\{c_i + c'_{k-i} : i \in \{0, \dots, k\}\}\,.$$
  $\rightsquigarrow$ For *most* d_k, the *largest* c_i and c'_j don't come into play!

Attack #1: Morphing products

- Observation:
  Tropical polynomial arithmetic is highly non-cancellable.

- Example: Let $F(x) := \bigoplus_i c_i \otimes x^{\otimes i}$ and $G(x) = \bigoplus_i c'_i \otimes x^{\otimes i}$.
  Then the n^{th} coefficient d_k of $F(x) \otimes G(x)$ looks like
  $$\min\{c_i + c'_{k-i} : i \in \{0, \ldots, k\}\}\,.$$
  $\rightsquigarrow$ For *most* d_k, the *largest* c_i and c'_j don't come into play!

- Attack:
  - Start from "trivial forgery" $(S_1, S_2) = (P \otimes M, P \otimes N)$.
  - Find positions i and j of S_1 and S_2 that can be changed (e.g., ± 1) without affecting the value of $S_1 \otimes S_2$.

Attack #1: Morphing products

```
U, V = one_v_poly(d, r), one_v_poly(d, r)
N = pol_times_pol2(U, V)
PN = pol_times_pol2(P, N)

rhs = pol_times_pol2(PM, PN)

for s,i in itertools.product((+1,-1), range(len(PM))):
    S1 = copy.deepcopy(PM)
    S1[i][0] += s
    if pol_times_pol2(S1, PN) == rhs:
        break

for s,i in itertools.product((+1,-1), range(len(PN))):
    S2 = copy.deepcopy(PN)
    S2[i][0] += s
    if pol_times_pol2(S1, S2) == rhs:
        break
```

Attack #2: Swapping divisors

- Observation: It is not necessary to fully factor M (or N).
- We already have $S_1 \otimes S_2 = P \otimes P \otimes M \otimes N$.
  Wanted: Some different factorization of this value.
  (satisfying constraints on degrees and coefficient sizes).

Attack #2: Swapping divisors

- Observation: It is not necessary to fully factor M (or N).
- We already have $S_1 \otimes S_2 = P \otimes P \otimes M \otimes N$.
  Wanted: Some different factorization of this value.
  (satisfying constraints on degrees and coefficient sizes).
- Attack:
  - Find equal-degree divisors D_1 of $P \otimes M$ and D_2 of $P \otimes N$.
  - *Swap them.*

Attack #2: Swapping divisors

- Observation: It is not necessary to fully factor M (or N).
- We already have $S_1 \otimes S_2 = P \otimes P \otimes M \otimes N$.
  Wanted: Some different factorization of this value.
  (satisfying constraints on degrees and coefficient sizes).
- Attack:
  - Find equal-degree divisors D_1 of $P \otimes M$ and D_2 of $P \otimes N$.
  - *Swap them.*
    In some more detail: Decompose $P \otimes M = D_1 \otimes R_1$ and $P \otimes N = D_2 \otimes R_2$.
    Then set $S_1 := D_1 \otimes R_2$ and $S_2 := D_2 \otimes R_1$.

Attack #2: Swapping divisors

- Observation: It is not necessary to fully factor M (or N).
- We already have $S_1 \otimes S_2 = P \otimes P \otimes M \otimes N$.
  Wanted: Some different factorization of this value.
  (satisfying constraints on degrees and coefficient sizes).
- Attack:
  - Find equal-degree divisors D_1 of $P \otimes M$ and D_2 of $P \otimes N$.
  - *Swap them.*
    In some more detail: Decompose $P \otimes M = D_1 \otimes R_1$ and $P \otimes N = D_2 \otimes R_2$.
    Then set $S_1 := D_1 \otimes R_2$ and $S_2 := D_2 \otimes R_1$.
- Finding (small-degree) divisors: Write $P \otimes M = D_1 \otimes R_1$ as a system of inequalities; feed them to a generic solver.
  I've had great success with the *z3* SMT solver.

Brown–Monico's attacks

- ePrint 2023/1837: Several new attack variants. 🎉
- One example: "double dividing".

Brown–Monico's attacks

- ePrint 2023/1837: Several new attack variants. 🎉
- One example: "double dividing".

Core idea: Tropical division of tropical polynomials.

- Defining property: $(F \oslash G) \otimes G = F$.
- Quotient does not always exist.
- However, $(F \otimes G) \oslash G$ always exists, but is usually $\neq F$.

Brown–Monico's attacks

- ePrint 2023/1837: Several new attack variants. 🎉
- One example: "double dividing".

Core idea: Tropical division of tropical polynomials.

- Defining property: $(F \oslash G) \otimes G = F$.
- Quotient does not always exist.
- However, $(F \otimes G) \oslash G$ always exists, but is usually $\neq F$.

<u>Attack</u>:

- Let $N \xleftarrow{\text{random}} T_{2d,2r}$.
- Set $S_1 = P \otimes M$ and $S_2 = P \otimes N$ and write $R := S_1 \otimes S_2$.

Brown–Monico's attacks

- ePrint 2023/1837: Several new attack variants. 🎉
- One example: "double dividing".

Core idea: Tropical division of tropical polynomials.

- Defining property: $(F \oslash G) \otimes G = F$.
- Quotient does not always exist.
- However, $(F \otimes G) \oslash G$ always exists, but is usually $\neq F$.

Attack:

- Let $N \xleftarrow{\text{random}} T_{2d,2r}$.
- Set $S_1 = P \otimes M$ and $S_2 = P \otimes N$ and write $R := S_1 \otimes S_2$.
- Compute $S_1' := R \oslash S_2$

Brown–Monico's attacks

- ePrint 2023/1837: Several new attack variants. 🎉
- One example: "double dividing".

Core idea: Tropical division of tropical polynomials.

- Defining property: $(F \oslash G) \otimes G = F$.
- Quotient does not always exist.
- However, $(F \otimes G) \oslash G$ always exists, but is usually $\neq F$.

Attack:

- Let $N \xleftarrow{\text{random}} T_{2d,2r}$.
- Set $S_1 = P \otimes M$ and $S_2 = P \otimes N$ and write $R := S_1 \otimes S_2$.
- Compute $S_1' := R \oslash S_2$ and subsequently $S_2' := R \oslash S_1'$.

Brown–Monico's attacks

- ePrint 2023/1837: Several new attack variants. 🎉
- One example: "double dividing".

Core idea: Tropical division of tropical polynomials.

- Defining property: $(F \oslash G) \otimes G = F$.
- Quotient does not always exist.
- However, $(F \otimes G) \oslash G$ always exists, but is usually $\neq F$.

Attack:

- Let $N \xleftarrow{\text{random}} T_{2d,2r}$.
- Set $S_1 = P \otimes M$ and $S_2 = P \otimes N$ and write $R := S_1 \otimes S_2$.
- Compute $S_1' := R \oslash S_2$ and subsequently $S_2' := R \oslash S_1'$.
- The forged signature is (S_1', S_2', N).

The updated ePrint (January 17, 2024)

- Degrees of X and Y are now distinct (U, V accordingly).
  I'm not sure what attack this is supposed to fix.

The updated ePrint (January 17, 2024)

- Degrees of X and Y are now distinct (U, V accordingly).
  I'm not sure what attack this is supposed to fix.

- First and last coefficients of X, Y are being forced to $0 \in \mathbb{T}$.
  $\rightsquigarrow$ irreducible a lot of the time $\rightsquigarrow$ finding small factors <u>allegedly</u> fails.

The updated ePrint (January 17, 2024)

- Degrees of X and Y are now distinct (U, V accordingly).
  I'm not sure what attack this is supposed to fix.
- First and last coefficients of X, Y are being forced to $0 \in \mathbb{T}$.
  $\rightsquigarrow$ irreducible a lot of the time $\rightsquigarrow$ finding small factors <u>allegedly</u> fails.
- Countermeasure from Brown–Monico:
  Check that $P \mid S_1, S_2$ and $M \nmid S_1, S_2$ and $N \nmid S_1, S_2$.
  Strangely, not included in updated ePrint 2023/1475.
  $\Longrightarrow$ "Double dividing" still works!

The updated ePrint (January 17, 2024)

- Degrees of X and Y are now distinct (U, V accordingly).
  I'm not sure what attack this is supposed to fix.
- First and last coefficients of X, Y are being forced to $0 \in \mathbb{T}$.
  $\rightsquigarrow$ irreducible a lot of the time $\rightsquigarrow$ finding small factors <u>allegedly</u> fails.
- Countermeasure from Brown–Monico:
  Check that $P \mid S_1, S_2$ and $M \nmid S_1, S_2$ and $N \nmid S_1, S_2$.
  Strangely, not included in updated ePrint 2023/1475.
  $\Longrightarrow$ "Double dividing" still works!
- (The "rehashing" attack from Brown–Monico also remains unfixed.)

김민순's attack

Yet another break, found while solving a CTF challenge:

`https://soon.haari.me/2023-christmas-ctf/#tropical-santa`

김민순's attack

Yet another break, found while solving a CTF challenge:

`https://soon.haari.me/2023-christmas-ctf/#tropical-santa`

<u>Attack</u>:

- Let $N \xleftarrow{\text{random}} T_{2d,2r}$.
- Set $S_1 = P \otimes M$ and $S_2 = P \otimes N$ and write $R = S_1 \otimes S_2$.
- Compute $S_1' := (R \oslash (P \otimes S_2)) \otimes P$.
- Compute $S_2' := (R \oslash (P \otimes S_1')) \otimes P$.
- The signature is (S_1', S_2', N).

김민순's attack

Yet another break, found while solving a CTF challenge:

https://soon.haari.me/2023-christmas-ctf/#tropical-santa

Attack:

- Let $N \xleftarrow{\text{random}} T_{2d,2r}$.
- Set $S_1 = P \otimes M$ and $S_2 = P \otimes N$ and write $R = S_1 \otimes S_2$.
- Compute $S_1' := (R \oslash (P \otimes S_2)) \otimes P$.
- Compute $S_2' := (R \oslash (P \otimes S_1')) \otimes P$.
- The signature is (S_1', S_2', N).

(Check: We have $S_1' \otimes S_2 = (R \oslash (P \otimes S_2)) \otimes P \otimes S_2 = R$. Similarly $S_1' \otimes S_2' = R$.)

김민순's attack

Yet another break, found while solving a CTF challenge:

`https://soon.haari.me/2023-christmas-ctf/#tropical-santa`

<u>Attack</u>:

- Let $N \xleftarrow{\text{random}} T_{2d,2r}$.
- Set $S_1 = P \otimes M$ and $S_2 = P \otimes N$ and write $R = S_1 \otimes S_2$.
- Compute $S_1' := (R \oslash (P \otimes S_2)) \otimes P$.
- Compute $S_2' := (R \oslash (P \otimes S_1')) \otimes P$.
- The signature is (S_1', S_2', N).

(Check: We have $S_1' \otimes S_2 = (R \oslash (P \otimes S_2)) \otimes P \otimes S_2 = R$. Similarly $S_1' \otimes S_2' = R$.)

!! This variant bypasses all proposed countermeasures. 

The updated ePrint (January 17, 2024)

- ...also proposes another new scheme to thwart the attacks.

The updated ePrint (January 17, 2024)

- ...also proposes another new scheme to thwart the attacks.
- It uses tropical multiplication and addition.

  Signature: (R, S, T, N, E) with some bounds.

  Verification:

$$P \otimes (R \oplus S) \oplus E \stackrel{?}{=} (P \otimes P) \oplus T$$

$$(R \otimes S) \oplus E \stackrel{?}{=} (P \otimes P) \oplus T \oplus (M \otimes N)$$

The updated ePrint (January 17, 2024)

- ...also proposes another new scheme to thwart the attacks.
- It uses tropical multiplication and addition.

  Signature: (R, S, T, N, E) with some bounds.

  Verification:

$$P \otimes (R \oplus S) \oplus E \stackrel{?}{=} (P \otimes P) \oplus T$$

$$(R \otimes S) \oplus E \stackrel{?}{=} (P \otimes P) \oplus T \oplus (M \otimes N)$$

- Stupid attack: Choose arbitrary R, S, N and set

$$T = E = \bigoplus_i \left(0 \otimes x^{\otimes i}\right).$$

  This validates for any message: Recall $\forall a \geq 0.\ a \oplus 0 = 0.$

This talk

How to break tropical signatures (in several different ways)

Some comments on cryptographic design methodology

Just bad luck?

Just bad luck?

- The “tropical signatures” construction combines two common what-I-argue-to-be-preventible-mistakes.

Just bad luck?

- The "tropical signatures" construction combines
  two common what-I-argue-to-be-preventible-mistakes.

(2) Focus on the **NP**-hardness of the underlying problem.

Just bad luck?

- The "tropical signatures" construction combines two common what-I-argue-to-be-preventible-mistakes.

(2) Focus on the **NP**-hardness of the underlying problem.

(1) Construction is not actually based on that problem.

(1) No reduction $\frown$

The tropical signatures paper argues that "security relies on [...] hardness of factoring one-variable tropical polynomials".

(1) No reduction $\ddot\frown$

The tropical signatures paper argues that “security relies on [...] hardness of factoring one-variable tropical polynomials”.

This is only true for *extremely* weak notions of “relies on”. Argument (paraphrased): If you can factor, it’s definitely dead. Essentially a case of “reduction in the wrong direction”!

(1) No reduction ☹

The tropical signatures paper argues that "security relies on [...] hardness of factoring one-variable tropical polynomials".

This is only true for *extremely* weak notions of "relies on". Argument (paraphrased): If you can factor, it's definitely dead. Essentially a case of "reduction in the wrong direction"!

<u>Case in point</u>:

- Breaking *any* public-key cryptosystem lies in **NP**, hence "is" an instance of an(y) **NP**-complete problem.

(1) No reduction ☹

The tropical signatures paper argues that "security relies on [...] hardness of factoring one-variable tropical polynomials".

This is only true for *extremely* weak notions of "relies on".
Argument (paraphrased): If you can factor, it's definitely dead.
Essentially a case of "reduction in the wrong direction"!

<u>Case in point</u>:

- Breaking *any* public-key cryptosystem lies in **NP**, hence "is" an instance of an(y) **NP**-complete problem.
- Stupid <u>example</u>: Rewrite SIKE in terms of binary circuits; now it is an instance of Circuit-SAT, which is **NP**-complete. Moreover, the only obvious way of attacking Circuit-SAT is to use a generic SAT solver, which cannot work because Circuit-SAT is **NP**-hard, so we're good!

(2) Cryptography *does not care* about **NP**-hardness

- By definition, **NP**-hardness is a worst-case notion.

(2) Cryptography *does not care* about **NP**-hardness

- By definition, **NP**-hardness is a worst-case notion.
- Cryptography needs random instances to be hard.

The big question:

(2) Cryptography *does not care* about **NP**-hardness

- By definition, **NP**-hardness is a worst-case notion.
- Cryptography needs random instances to be hard.

The big question:

> Are we actually sampling the hard instances?

(2) Cryptography *does not care* about **NP**-hardness

- By definition, **NP**-hardness is a worst-case notion.
- Cryptography needs random instances to be hard.

The big question:

> **Are we actually sampling the <u>hard instances</u>?**

- (Answer for tropical signatures: It does not seem so ⌣̈.)

(2) Cryptography *does not care* about **NP**-hardness

- By definition, **NP**-hardness is a worst-case notion.
- Cryptography needs random instances to be hard.

The big question:

> **Are we actually sampling the <u>hard instances</u>?**

- (Answer for tropical signatures: It does not seem so ☺.)
- This is what average-case hardness is about.

(2) Cryptography *does not care* about **NP**-hardness

A Hard Problem That is Almost Always Easy

George Havas and B.S. Majewski

Key Centre for Software Technology, Department of Computer Science, University of Queensland, Queensland 4072, Australia

Abstract. NP-completeness is, in a well-defined sense, a worst case notion. Thus, 3–colorability of a graph, for a randomly generated graph, can be determined in *constant expected time* even though the general problem is NP-complete. The reason for this is that some hard problems exhibit a *structure where only a small (perhaps exponentially small)* fraction of all possible instances is intractable, while the remaining large fraction has a polynomial time solution algorithm.

(2) Cryptography *does not care* about **NP**-hardness

- $\nexists$ cryptosystem which is known to be **NP**-hard to break.
  (In fact, there exist arguments that cryptography from **NP**-hard problems may be impossible for fundamental reasons.)

(2) Cryptography *does not care* about **NP**-hardness

- $\nexists$ cryptosystem which is known to be **NP**-hard to break.
  (In fact, there exist arguments that cryptography from **NP**-hard problems may be impossible for fundamental reasons.)

Papadimitriou, 1995: It is now common knowledge among computer scientists that NP-completeness is largely irrelevant to public-key cryptography, since in that area one needs sophisticated *cryptographic assumptions* that go beyond NP-completeness and worst-case polynomial-time computation [19]; furthermore, cryptographic protocols based on NP-complete problems have been ill-fated.

Questions?

lorenz@yx7.cc