Applications of fast integer and polynomial lattice reduction in cryptography

Nadia Heninger

UC San Diego

September 28, 2023

Are you looking for fun cryptographic applications?

NIST Post-Quantum Round 1 Additional Signatures

family	submissions
code-based	6
isogenies	1
lattice-based	7
MPC-in-the-head	7
multivariate	10
symmetric	4
other	5

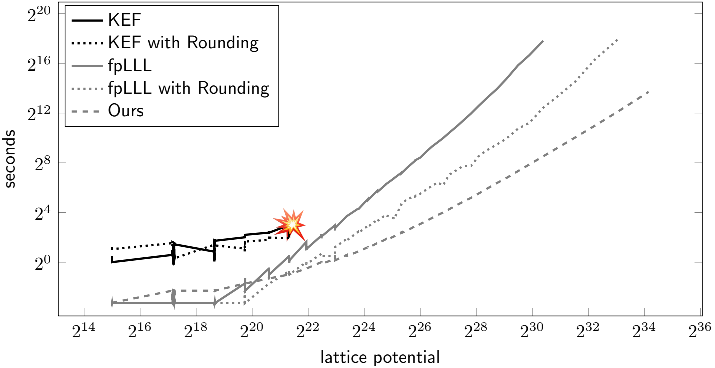
Are you looking for fun cryptographic applications?

NIST Post-Quantum Round 1 Additional Signatures

family	submissions
code-based	6
isogenies	1
lattice-based	7
MPC-in-the-head	7
multivariate	10
symmetric	4
other	5

Are you looking for fast lattice reduction?

Coppersmith RSA small public exponent attack



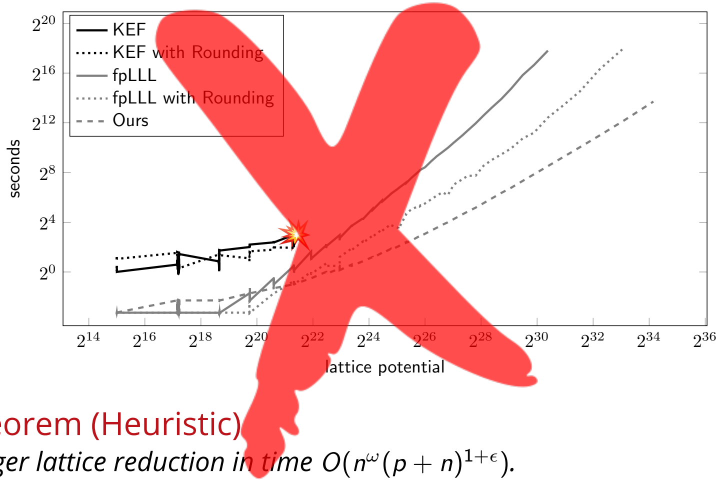
Theorem (Heuristic)

Integer lattice reduction in time $O(n^{\omega}(p+n)^{1+\epsilon})$.

"Fast Practical Lattice Reduction through Iterated Compression"
Keegan Ryan and Nadia Heninger *Crypto 2023*
`https://github.com/keeganryan/flatter`

Are you looking for fast lattice reduction?

Coppersmith RSA small public exponent attack



Theorem (Heuristic)

Integer lattice reduction in time $O(n^{\omega}(p+n)^{1+\epsilon})$.

"Fast Practical Lattice Reduction through Iterated Compression"
Keegan Ryan and Nadia Heninger *Crypto 2023*
`https://github.com/keeganryan/flatter`

This talk

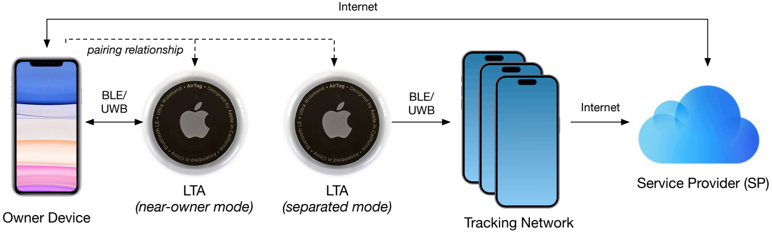
1. Privacy-preserving Airtag stalker detection (with polynomial lattice reduction!)

2. On the Possibility of a Backdoor in the Micali-Schnorr Generator (with integer lattices!)

Privacy-preserving Airtag stalker detection

Abuse-Resistant Location Tracking: Balancing Privacy and Safety in the Offline Finding Ecosystem. Gabrielle Beck, Harry Eldridge, Matthew Green, Nadia Heninger, and Abhishek Jain. `https://eprint.iacr.org/2023/1332`

How do airtags work?



1. Airtags emit a 248-bit Bluetooth Low Energy broadcast every 2s.
2. Any nearby devices receive broadcasts, collect, and upload to Apple's servers along with location.
3. Users can query server for tag identifier and receive location reports.

Privacy threats

Threat: Airtags allow others to monitor your location

- Countermeasure: Tags rotate identifiers periodically.

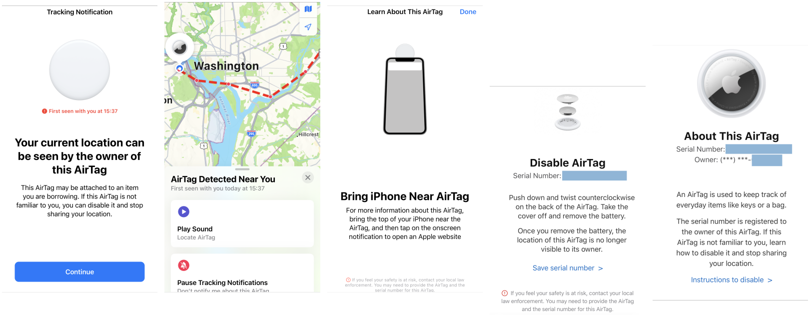
Privacy threats

Threat: Airtags allow others to monitor your location

- Countermeasure: Tags rotate identifiers periodically.

Threat: Stalker places an Airtag on your car/person

- Countermeasure: Your device sends an alert if it sees the same identifier for some period of time.



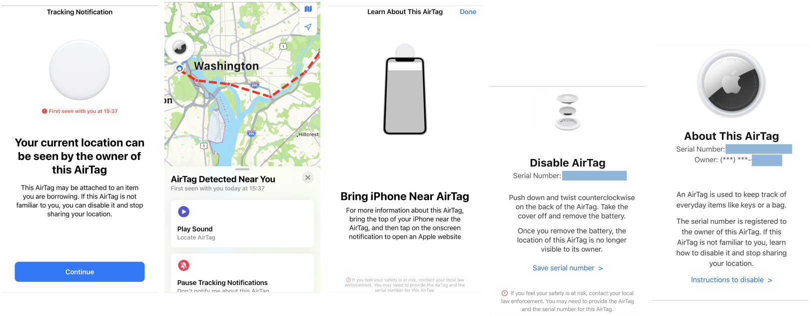
Privacy threats

Threat: Airtags allow others to monitor your location

- Countermeasure: Tags rotate identifiers periodically.

Threat: Stalker places an Airtag on your car/person

- Countermeasure: Your device sends an alert if it sees the same identifier for some period of time.



Research Goal

Allow stalker detection while maximizing privacy against location tracking.

Construction Idea:

- Use Shamir secret sharing.
- Tag chooses secret polynomial $f \in \mathbb{F}_q[z]$ and broadcasts evaluations

$$(z_1, f(z_1)), (z_2, f(z_2)), \ldots, (z_n, f(z_n))$$

- Privacy threshold: No party observing fewer than $\deg f$ broadcasts can distinguish from random.
- Noise: A moving device will receive broadcasts from many tags; some broadcasts dropped.

Stalker detection is polynomial interpolation with noise
= Reed-Solomon decoding.

Detecting multiple stalkers = Reed-Solomon list decoding

Polynomial interpolation with noise

Input:

z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8
y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8
$r_1(z_1)$	$r_1(z_2)$	$r_2(z_3)$	$r_1(z_4)$	$r_3(z_5)$	$r_1(z_6)$	$r_2(z_7)$	$r_1(z_8)$

Desired output: r_1

Let

$$N(z) = \prod_i (z - z_i); \quad H(z) = \prod_{i \mid r_1(z_i) = y_i} (z - z_i)$$

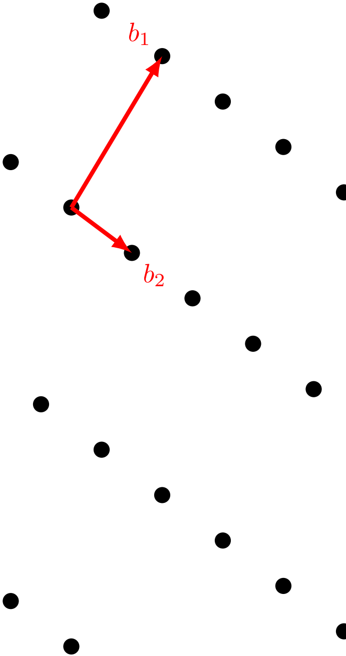
Interpolate $a(z)$ so

$$a(z_i) = y_i$$

Then

$$\gcd(r_1(z) - a(z), N(z)) = H(z)$$

Polynomial lattices



Definition

$\mathbb{F}[z]$**-module**: $B = (b_1, b_2, \ldots, b_n)$, $b_i \in \mathbb{F}(z)^n$

$L(B) = \{v_i | v_i = \sum_i a_i b_i, a_i \in \mathbb{F}[z], b_i \in B\}$.

Definition

Vector **length**

$\deg v = \max_i \deg v_i$.

Definition

Determinant

$\det L(B) = \det B$

Lattice basis reduction for polynomial lattices

von zur Gathen, Mulders and Storjohann

Pivot: right-most element of maximal degree in vector

Definition

A basis is **reduced** if its pivots are all in different columns.

Fact

If $\{b_i\}$ *is a reduced basis for* L*,* $\deg\det L = \sum_i \deg b_i$*.*

Theorem

A reduced basis contains a vector with $\deg v < (\deg\det L)/\dim L$*.*

Theorem

A reduced basis contains a shortest vector of L*.*

Theorem (Giorgi, Jeannerod, Villard)

$(\dim L)^{\omega+o(1)}D$ *running time for polynomial lattice reduction*

($D =$*max degree)*

Reed-Solomon decoding via polynomial lattices

Input:

z_1	z_2	z_3	z_4	z_5	z_6	z_7	$\cdots$	z_n
y_1	y_2	y_3	y_4	y_5	y_6	y_7	$\cdots$	y_n

Output: r s.t. $\deg r \leq \ell$ and $r(x_i) = y_i$ for $\geq h$ values of i.

1. Let
$$N(z) = \prod_i (z - z_i); \quad a(z) \mid a(z_i) = y_i \; \forall i$$
2. Construct
$$B = \begin{bmatrix} z^\ell & -a(z) \\ & N(z) \end{bmatrix}$$
$$\dim L(B) = 2$$
$$\deg \det L(B) = \ell + n$$
3. Reduce B to find a vector $(z^\ell q_1(z), q_2(z))$.
4. If $(\ell + n)/2 < h$ then solution $r(z) = q_2(z)/q_1(z)$.

List-decoding for Reed-Solomon codes

Theorem (Guruswami Sudan)

In polynomial time can find all $r_i(z)$ s.t. $\deg r_i(z) < h^2/n$.

Previous construction: Construct polynomial $Q(x) = q_1(z)x + q_2(z)$ with the property that $Q(r(z)) = 0$.

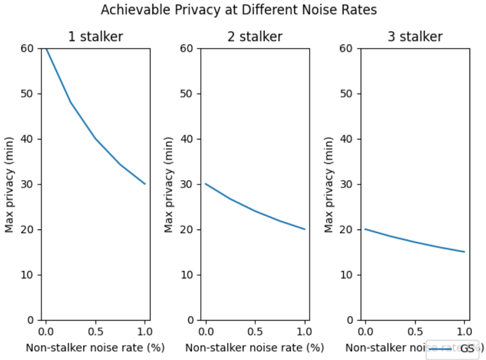
Guruswami-Sudan construction: Construct polynomial $Q(x)$ of degree t; roots r_i are among t roots of Q.

Theorem (Jeannerod, Neiger, Schost, Villard)

Interpolation/reduction step can be done in $O(t^{\omega-1}\,\mathsf{M}(t\ell)\log(t\ell)\log(\ell))$ *time.*

Applying Guruswami-Sudan for stalker detection

Problem 1: Privacy threshold far from stalker detection time.



Problem 2: Huge memory consumption and running time at asymptotic bounds.
Dimension $t \approx hn$ and degree nh^2; $n > 1800$ for 2s broadcasts and 1h window.

Alternative coding-based constructions

Various extensions of Reed-Solomon codes have better theoretical decoding rates:

- Parvaresh-Vardy
- Folded Reed-Solomon Codes

However:

- Algebraically structured correlations within/across broadcasts may not satisfy secret sharing properties. **Open problem:** Say something more rigorous about this.

- Don’t perform well for our desired parameters.

Construction: Multiple polynomial evalutions

(Like an interleaved Reed-Solomon code)

In each epoch, user generates random $r_1, \ldots r_c \in \mathbb{F}[z]$.

$$\begin{array}{|c|cccccccc}
\hline
z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 \\
r_1(z_1) & r_1(z_2) & r_1(z_3) & r_1(z_4) & r_1(z_5) & r_1(z_6) & r_1(z_7) & r_1(z_8) \\
r_2(z_1) & r_2(z_2) & r_2(z_3) & r_2(z_4) & r_2(z_5) & r_2(z_6) & r_2(z_7) & r_2(z_8) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
r_c(z_1) & r_c(z_2) & r_c(z_3) & r_c(z_4) & r_c(z_5) & r_c(z_6) & r_c(z_7) & r_c(z_8) \\
\hline
\end{array}$$

Noisy simultaneous polynomial recovery

Input:

$$\begin{array}{cccccccc}
z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 \\
r_{11}(z_1) & r_{11}(z_2) & r_{21}(z_3) & r_{11}(z_4) & r_{31}(z_5) & r_{21}(z_6) & r_{21}(z_7) & r_{11}(z_8) \\
r_{12}(z_1) & r_{12}(z_2) & r_{22}(z_3) & r_{12}(z_4) & r_{32}(z_5) & r_{22}(z_6) & r_{22}(z_7) & r_{12}(z_8) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
r_{1c}(z_1) & r_{1c}(z_2) & r_{2c}(z_3) & r_{1c}(z_4) & r_{3c}(z_5) & r_{2c}(z_6) & r_{2c}(z_7) & r_{1c}(z_8)
\end{array}$$

Desired output: $r_{11}, r_{12}, \ldots, r_{1c}$, maybe $r_{21}, r_{22}, \ldots, r_{2c}$

Noisy simultaneous polynomial recovery

Input:

z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8
$r_{11}(z_1)$	$r_{11}(z_2)$	$r_{21}(z_3)$	$r_{11}(z_4)$	$r_{31}(z_5)$	$r_{21}(z_6)$	$r_{21}(z_7)$	$r_{11}(z_8)$
$r_{12}(z_1)$	$r_{12}(z_2)$	$r_{22}(z_3)$	$r_{12}(z_4)$	$r_{32}(z_5)$	$r_{22}(z_6)$	$r_{22}(z_7)$	$r_{12}(z_8)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$r_{1c}(z_1)$	$r_{1c}(z_2)$	$r_{2c}(z_3)$	$r_{1c}(z_4)$	$r_{3c}(z_5)$	$r_{2c}(z_6)$	$r_{2c}(z_7)$	$r_{1c}(z_8)$

Desired output: $r_{11}, r_{12}, \ldots, r_{1c}$

Let

$$N(z) = \prod_i (z - z_i); \quad H(z) = \prod_{i \mid r_{1j}(z_i) = y_{ij} \forall j} (z - z_i)$$

Interpolate $a_1(z), \ldots, a_c(z)$ so $a_j(z_i) = y_{ij} \, \forall i, j$

Then

$$\gcd(r_{11}(z) - a_1(z), r_{12}(z) - a_2(z), \ldots, N(z)) = H(z)$$

Noisy simultaneous polynomial recovery via lattices

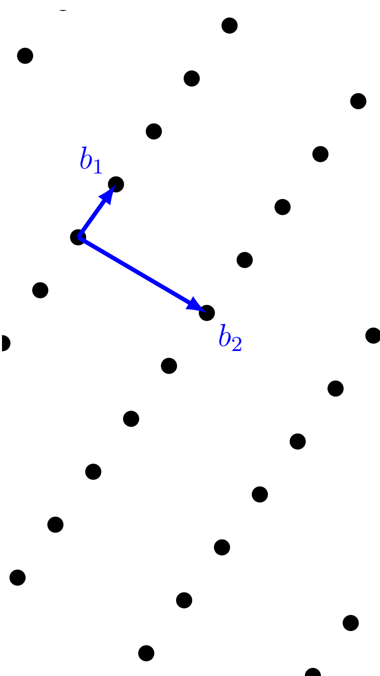
1. Let $N(z) = \prod_i (z - z_i)$; $a_1(z), \ldots, a_c(z) \mid a_j(z_i) = y_{ij}$
2. Construct

$$B = \begin{bmatrix} z^\ell & & & & -a_1(z) \\ & z^\ell & & & -a_2(z) \\ & & \ddots & & \vdots \\ & & & z^\ell & -a_c(z) \\ & & & & N(z) \end{bmatrix}$$

$$\dim L(B) = c + 1$$
$$\deg \det L(B) = c\ell + n$$

3. Reduce B to find m short vectors.
4. Map vectors to linear equations in m unkowns; solve system for r_{ij}.
5. If $(c\ell + n)/(c + 1) < h$ then hope to find solution.

Polynomial lattice duality



Definition

The *dual* lattice L^* is defined as all vectors $w \in \mathbb{F}(x)^m$ satisfying $\langle w, v \rangle \in \mathbb{F}[x]$ for $v \in L$.

- $(L^*)^* = L$

Explicit basis: If B is full rank, then $(B^{-1})^T$ is an explicit basis for $L^*(B)$.

Noisy simultaneous polynomial recovery, dual form

1. Let $N(z) = \prod_i (z - z_i)$; $a_1(z), \ldots, a_c(z) \mid a_j(z_i) = y_{ij}$
2. Construct rescaled (by $z^\ell N(z)$) dual basis:

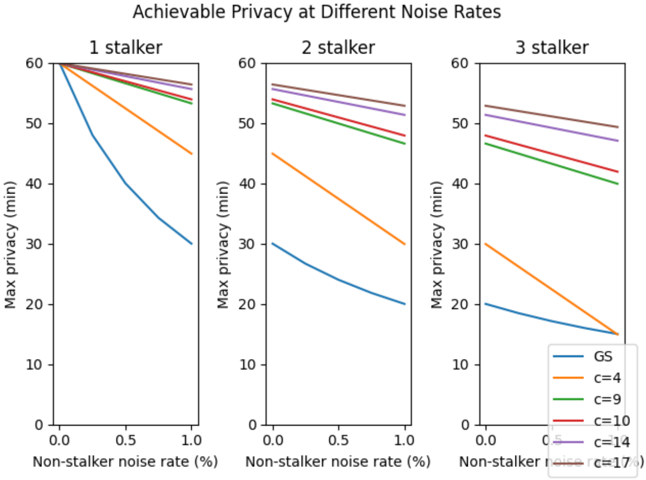
$$B^* = \begin{bmatrix} N(z) & & & & \\ & N(z) & & & \\ & & \ddots & & \\ & & & N(z) & \\ a_1(z) & a_2(z) & \ldots & a_c(z) & z^\ell \end{bmatrix}$$

$$\dim L(B^*) = c + 1$$
$$\deg \det L(B^*) = \ell + cn$$

3. Reduce B^*.
4. Let $E(z) = \prod_{i \text{ error}} (z - z_i)$. Target vector $v = (r_1(z)E(z), r_2(z)E(z), \ldots, r_c(z)E(z), x^\ell E(z))$ in L^* by construction.
5. If $(\ell + cn)/(c+1) > n - h + \ell$ then expect v to be shortest vector. (Equivalent to bound obtained by primal.)

Multi-polynomial recovery for stalker detection

Results: Privacy threshold improves with more curves.



Practical considerations: BLE broadcasts have 246 bits available. So e.g. for $c = 10$ can use 22-bit $\mathbb{F}$.

Dealing with multiple stalkers/valid solutions

Semi-principled approach: Use higher degree polynomials.

- Impractical: Dimension increases exponentially with degree.
- Remains heuristic.

Dealing with multiple stalkers/valid solutions

Ad hoc approach with "linear" construction:
If multiple valid solutions match same number of inputs:

- Reduced lattice basis contains multiple vectors matching target length.
- Vectors contain arbitrary-looking rational functions.
- We have a sort of ad hoc construction to recover the targets after another reduction.

If one valid solution matches ≥ 2 more points than others:

- The most matchiest one is in the reduced basis; the others are not.
- We can remove the matching points and iterate to recover the others.

Open question: What is going on here?

Thoughts/Discussion

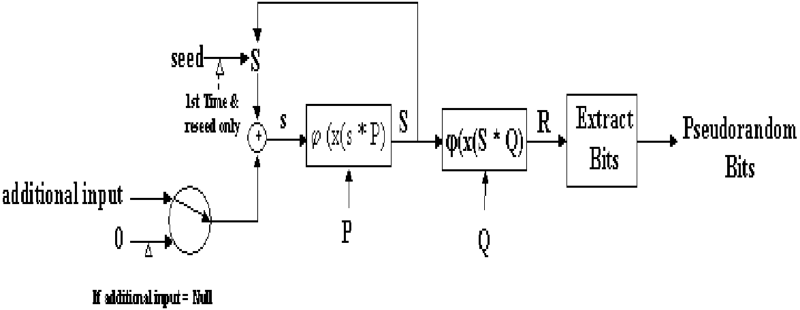
- Nearly all papers in this area focus on asymptotics; application-oriented readers have great difficulty setting or extracting actual parameters and running times.
- **Open question:** More formal/less heuristic theorems matching case where received shares/messages are all polynomial evaluations and not random noise.
- I am increasingly persuaded that what I presented as the "dual" form is the "correct" formulation for all these types of problems (including integer versions for multivariate Coppersmith-type methods).
- Cryptographic secret-sharing community has been applying some of this stuff for years in often exotic settings.

On the possibility of a backdoor in the Micali-Schnorr generator

On the Possibility of a Backdoor in the Micali-Schnorr Generator. Hannah Davis, Matthew D. Green, Nadia Heninger, Keegan Ryan, and Adam Suhl.
`https://eprint.iacr.org/2023/440`

2004: Dual EC presented at NIST workshop

ECC DRBG Flowchart



2005-2006: Dual EC standardized in NIST SP 800-90A

A.1 Constants for the Dual_EC_DRBG

The **Dual_EC_DRBG** requires the specifications of an elliptic curve and two points on the elliptic curve. One of the following NIST **approved** curves with associated points **shall** be used in applications requiring certification under [FIPS 140]. More details about these curves may be found in [FIPS 186]. If alternative points are desired, they **shall** be generated as specified in Appendix A.2.

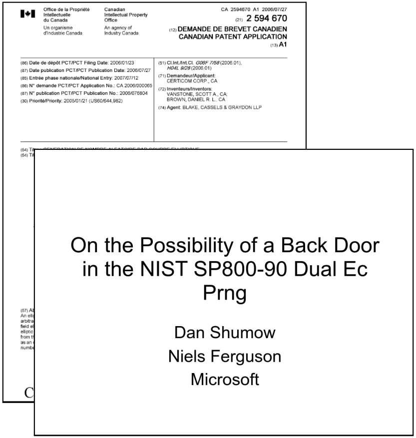
```
Px = 6b17d1f2 e12c4247 f8bce6e5 63a440f2 77037d81 2deb33a0
     f4a13945 d898c296

Py = 4fe342e2 fe1a7f9b 8ee7eb4a 7c0f9e16 2bce3357 6b315ece
     cbb64068 37bf51f5

Qx = c97445f4 5cdef9f0 d3e05e1e 585fc297 235b82b5 be8ff3ef
     ca67c598 52018192

Qy = b28ef557 ba31dfcb dd21ac46 e2a91e3c 304f44cb 87058ada
     2cb81515 1e610046
```

2005-2007: State-recovery backdoor possible in Dual EC



Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2594670 A1 2006/07/27

(21) 2 594 670

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(86) Date de dépôt PCT/PCT Filing Date: 2006/01/23
(87) Date publication PCT/PCT Publication Date: 2006/07/27
(85) Entrée phase nationale/National Entry: 2007/07/12
(86) N° demande PCT/PCT Application No.: CA 2006/000065
(87) N° publication PCT/PCT Publication No.: 2006/076804
(30) Priorité/Priority: 2005/01/21 (US60/644,982)

(51) Cl.Int./Int.Cl. *G06F 7/58* (2006.01), *H04L 9/28* (2006.01)
(71) Demandeur/Applicant: CERTICOM CORP., CA
(72) Inventeurs/Inventors: VANSTONE, SCOTT A., CA; BROWN, DANIEL R. L., CA
(74) Agent: BLAKE, CASSELS & GRAYDON LLP

On the Possibility of a Back Door in the NIST SP800-90 Dual Ec Prng

Dan Shumow
Niels Ferguson
Microsoft

“The relationship between P and Q [in Dual EC] is used as an escrow key and stored…the output of the generator [is used] to reconstruct the random number with the escrow key."

2012-2015: Hack of Juniper Network's Dual EC constants

Important Announcement about ScreenOS®

By dscholl posted 12-17-2015 09:02

1 Recommend

IMPORTANT JUNIPER SECURITY ANNOUNCEMENT

CUSTOMER UPDATE: DECEMBER 20, 2015

Administrative Access (CVE-2015-7755) only affects ScreenOS 6.3.0r17 through 6.3.0r20. VPN Decryption (CVE-2015-7756) only affects ScreenOS 6.2.0r15 through 6.2.0r18 and 6.3.0r12 through 6.3.0r20.

We strongly recommend that all customers update their systems and apply these patched releases with the highest priority.

POSTED BY BOB WORRALL, SVP CHIEF INFORMATION OFFICER ON DECEMBER 17, 2015

2012-2015: Hack of Juniper Network's Dual EC constants

Important Announcement about ScreenOS®

By dscholl posted 12-17-2015 09:02

1 Recommend

IMPORTANT JUNIPER SECURITY ANNOUNCEMENT

CUSTOMER UPDATE: DECEMBER 20, 2015

Administrative Access (CVE-2015-7755) only affects ScreenOS 6.3.0r17 through 6.3.0r20. VPN Decryption (CVE-2015-77 through 6.3.0r20.

We strongly recommend that all customers update their systems and apply these patched releases with the highest priority.

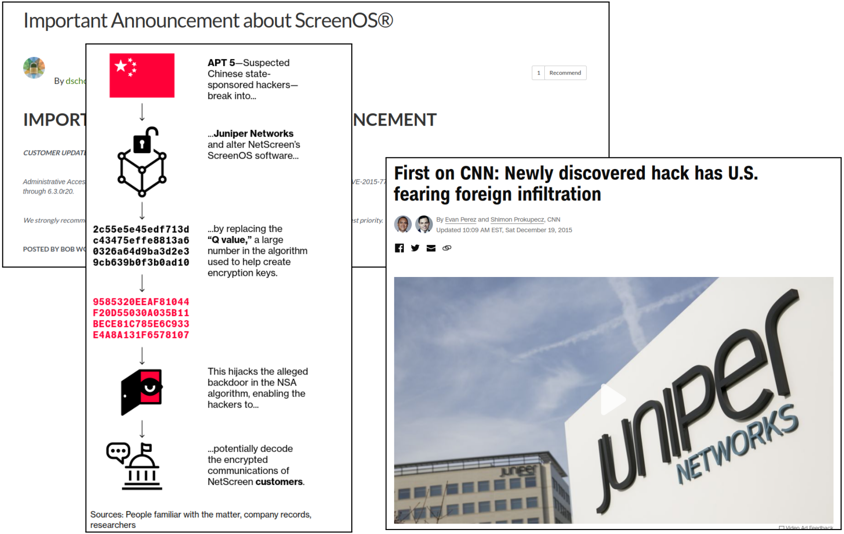
POSTED BY BOB WORRALL, SVP CHIEF INFORMATION OFFICER ON DECEMBER 17, 2015

First on CNN: Newly discovered hack has U.S. fearing foreign infiltration

By Evan Perez and Shimon Prokupecz, CNN
Updated 10:09 AM EST, Sat December 19, 2015

Video Ad Feedback

2012-2015: Hack of Juniper Network's Dual EC constants



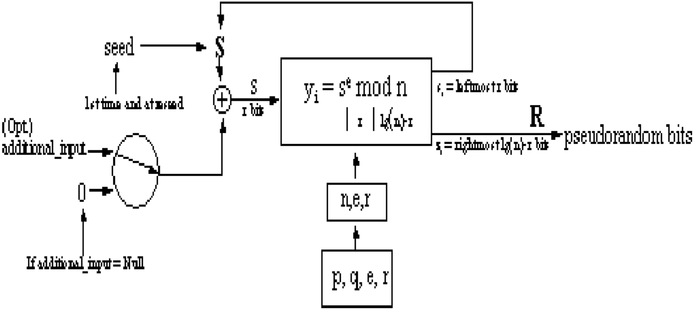
History of Dual EC

Dual EC

2004	Proposed inclusion in ANSI x9.82
2005	NIST SP 800-9A draft
2005-2007	Identification of possible backdoor
2013	Snowden Disclosures
2014	Removal from SP 800-90A
2012-2015	Exploitation of Juniper Networks

2004: Micali-Schnorr presented at NIST workshop

Micali-Schnorr DRBG



2005: Micali-Schnorr standardized in ISO 18031

Each modulus is of the form $n = pq$ with $p = 2p_1 + 1$, $q = 2q_1 + 1$, where p_1 and q_1 are $(\lg(n)/2 - 1)$-bit primes.

D.2.2 Default modulus *n* of size 1024 bits

The hexadecimal value of the modulus n is:

```
b66fbfda fbac2fd8 2eb13dc4 4fa170ff c9f7c7b5 1d55b214 4cc2257b 29df3f62
b421b158 0753f304 a671ff8b 55dd8abf b53d31ab a0ad742f 21857acf 814af3f1
e126d771 a61eca54 e62bfdb5 85c311b0 58e9cd3f aab758a5 e2896849 6ec1dd51
d0355aa1 55d4d912 6140dcfa b9b03f62 a5032d06 536d8574 0988f384 27f35885
```

D.2.3 Default modulus *n* of size 2048 bits

The hexadecimal value of the modulus n is:

```
c11a01f2 5daf396a a927157b af6f504f 78cba324 57b58c6b f7d851af 42385cc7
905b06f4 1f6d47ab 1b3a2c12 17d14d15 070c9da5 24734ada 2fe17a95 e600ae9a
4f8b1a66 96661e40 7d3043ec d1023126 5d8ea0d1 81cf23c6 dd3dec9e b3fce204
5b9299bb cca63dee 435a2251 ad0765d4 9d29db2e f5aba161 279aeb5f 6899fe48
7973e36c 1fb13086 d9231b6b 925a8495 4ba0fbca fea844ea 77a9f852 f86915a4
e71bd0ba b9b269c3 9a7a827a 41311ffa 4470140c 8b6509fe 5dbd39e3 ec816066
2d036e13 0e07e233 06a39b18 db0e8efe 64418880 81ac3673 2b4091f6 63690d03
3b486d74 371a20fc 3e214bce 7ed0e797 5ea44453 cd161d32 e8185204 59896571
```

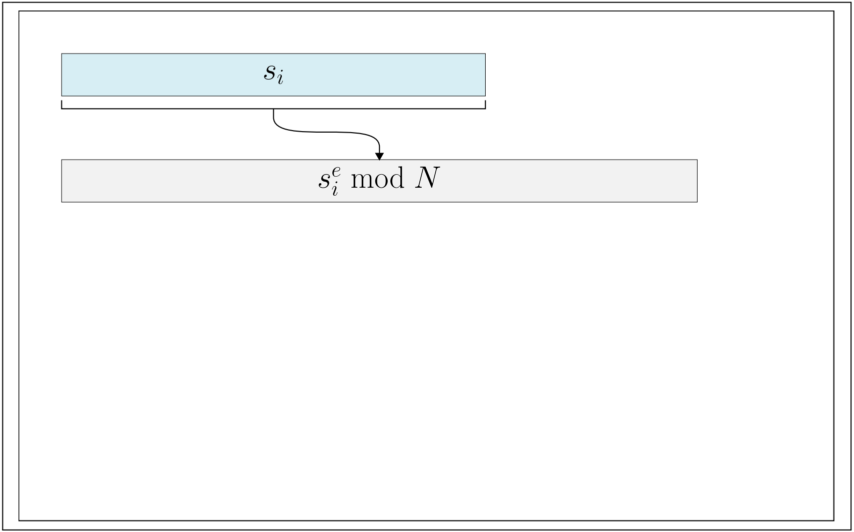
History of Dual EC…and Micali-Schnorr

Dual EC		Micali-Schnorr
2004	Proposed inclusion in ANSI x9.82	✓
2005	NIST SP 800-9A draft	ISO 18031
2005-2007	Identification of possible backdoor	?
2013	Snowden Disclosures	
2014	Removal from SP 800-90A	
2012-2015	Exploitation of Juniper Networks	

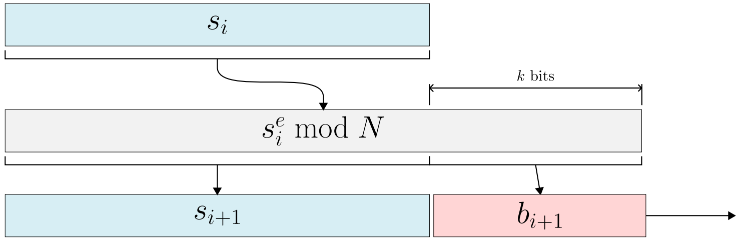
Micali-Schnorr's design: repeated RSA encryption

s_i

Micali-Schnorr's design: repeated RSA encryption

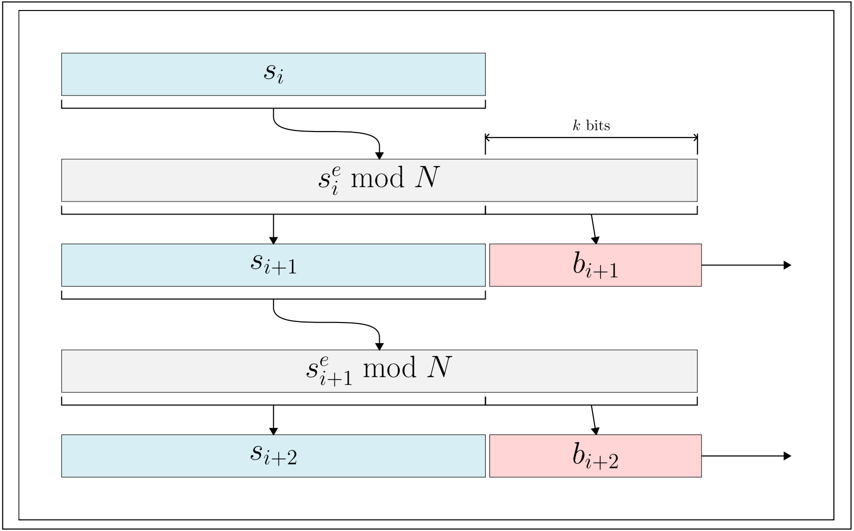


Micali-Schnorr's design: repeated RSA encryption



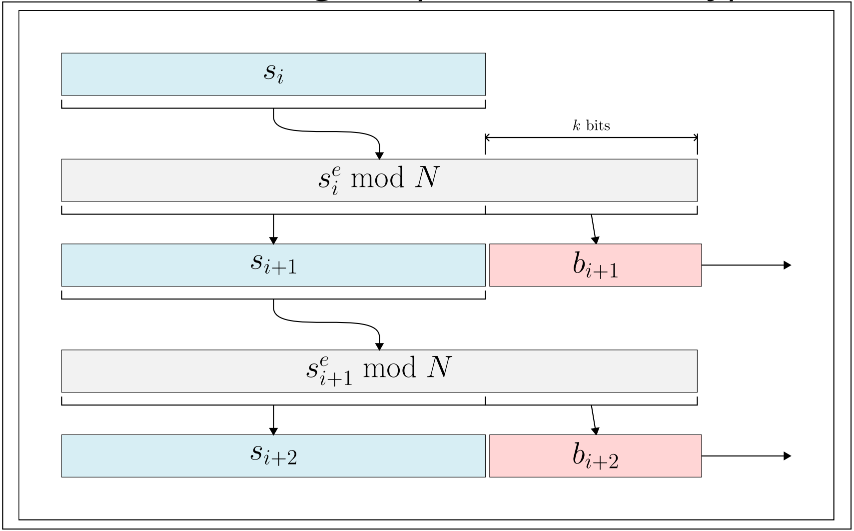
$$2^k s_{i+1} + b_{i+1} \equiv s_i{}^e \pmod{N}$$

Micali-Schnorr's design: repeated RSA encryption



$$2^k s_{i+1} + b_{i+1} \equiv s_i{}^e \pmod{N}$$

Micali-Schnorr's design: repeated RSA encryption



Unclear how to recover the state using RSA decryption.

Does the factorization of the public modulus lead to an attack against Micali-Schnorr?

Does the factorization, or otherwise malicious construction, of the public modulus lead to an attack against Micali-Schnorr?

Attacks against pseudorandom number generators

Attack model: Adversary knows or controls all parameters except for initial seed, and observes algorithm outputs.

Attacker would like to:

- Compute current secret state.
- Predict future outputs.
- Distinguish outputs from truly random values.

Observation 1

There is no *simple* backdoor in Micali-Schnorr.

No simple backdoors in Micali-Schnorr

Theorem

If RSA encryption is replaced with an invertible random function then the Micali-Schnorr construction is provably secure.

Corollary

Any potential backdoor in Micali-Schnorr must exploit the non-random structure of textbook RSA encryption.

RSA decryption alone is not enough.

No simple backdoors in Micali-Schnorr

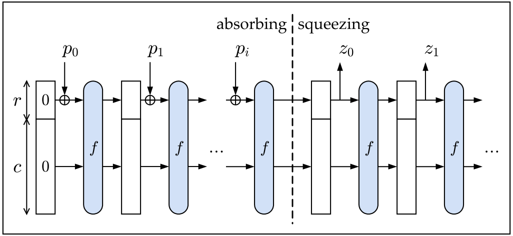
Theorem

If RSA encryption is replaced with an invertible random function then the Micali-Schnorr construction is provably secure.

Corollary

Any potential backdoor in Micali-Schnorr must exploit the non-random structure of textbook RSA encryption.

RSA decryption alone is not enough.



Micali-Schnorr is like a sponge with duplex construction.

Observation 2

There is an algebraic attack on the standard with non-default settings

Attempting Coppersmith-type methods

We want to recover unknown state from observed output.

$$s_0{}^e - 2^k s_1 - b_1 \equiv 0 \bmod N$$
$$s_1{}^e - 2^k s_2 - b_2 \equiv 0 \bmod N$$

Attempting Coppersmith-type methods

We want to recover unknown state from observed output.

$$s_0{}^e - 2^k s_1 - b_1 \equiv 0 \bmod N$$

$$s_1{}^e - 2^k s_2 - b_2 \equiv 0 \bmod N$$

Let $|s_i| < R = 2^r$. Construct the lattice basis

$$B = \begin{bmatrix} R^e & 0 & -2^k R & 0 & -b_1 \\ 0 & R^e & 0 & -2^k R & -b_2 \\ 0 & 0 & NR & 0 & 0 \\ 0 & 0 & 0 & NR & 0 \\ 0 & 0 & 0 & 0 & N \end{bmatrix}$$

$$\det L(B) = R^{2e+2} N^3$$
$$\dim L(B) = 5$$

Success condition (ignoring small constants):

$$(\det L(B))^{1/\dim L(B)} = (R^{2e+2} N^3)^{1/5} < N$$

This gives $R < N^{1/(e+1)}$ or $r < n/(e+1)$.

Attempting Coppersmith-type methods

We want to recover unknown state from observed output.

$$s_0{}^e - 2^k s_1 - b_1 \equiv 0 \bmod N$$
$$s_1{}^e - 2^k s_2 - b_2 \equiv 0 \bmod N$$

Let $|s_i| < R = 2^r$. Construct the lattice basis

$$B = \begin{bmatrix} R^e & 0 & -2^k R & 0 & -b_1 \\ 0 & R^e & 0 & -2^k R & -b_2 \\ 0 & 0 & NR & 0 & 0 \\ 0 & 0 & 0 & NR & 0 \\ 0 & 0 & 0 & 0 & N \end{bmatrix}$$

$$\det L(B) = R^{2e+2} N^3$$
$$\dim L(B) = 5$$

Success condition (ignoring small constants):

$$(\det L(B))^{1/\dim L(B)} = (R^{2e+2} N^3)^{1/5} < N$$

This gives $R < N^{1/(e+1)}$ or $r < n/(e+1)$.
Doesn't work. ISO 18031 sets $r = 2n/e$.

Backdooring Micali-Schnorr with non-default exponent

Backdoor idea: Use non-default public exponent e where the *private exponent* d is small. (e.g. $e = 3^{-1} \bmod \varphi N$)

Coppersmith's method successfully solves this polynomial.

$$(s_{i+1}2^k + b_{i+1})^d \equiv s_i \bmod N$$

Backdooring Micali-Schnorr with non-default exponent

Backdoor idea: Use non-default public exponent e where the *private exponent* d is small. (e.g. $e = 3^{-1} \bmod \varphi N$)

Coppersmith's method successfully solves this polynomial.

$$(s_{i+1} 2^k + b_{i+1})^d \equiv s_i \bmod N$$

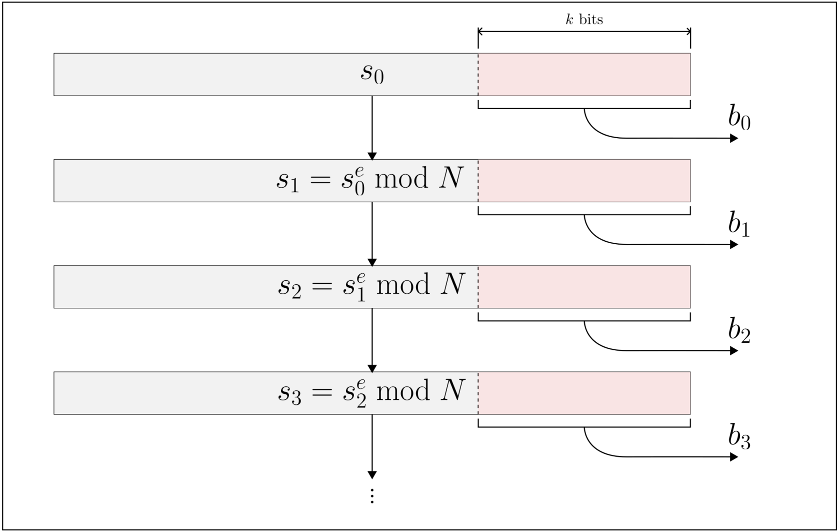
$\boxed{\text{ISO 18031: “The implementation should allow” non-default } e.}$

But this is not a satisfying backdoor: Large e looks suspicious.

Observation 3

We can force short cycles in a *related* RSA-based construction

RSA PRG



- State $s_i = {s_0}^{e^i} \bmod N$

RSA PRG can have short cycles

RSA PRG with $N = 5154904286740261$ and $e = 3$.

Iteration	Value	State s_i	Output b_i
0	s_0	4047975530247052	`338c`
1	${s_0}^e$	2492861700191393	`34a1`
2	${s_0}^{e^2}$	4862773567328857	`9259`
...	...	...	...
16	${s_0}^{e^{16}}$	810645248255668	`a6b4`
17	${s_0}^{e^{17}}$	2887166220613321	`b6c9`
18	${s_0}^{e^{18}}$	3479941204398616	`d218`

RSA PRG can have short cycles

RSA PRG with $N = 5154904286740261$ and $e = 3$.

Iteration	Value	State s_i	Output b_i
0	s_0	4047975530247052	`338c`
1	${s_0}^e$	2492861700191393	`34a1`
2	${s_0}^{e^2}$	4862773567328857	`9259`
...	...	...	...
16	${s_0}^{e^{16}}$	810645248255668	`a6b4`
17	${s_0}^{e^{17}}$	2887166220613321	`b6c9`
18	${s_0}^{e^{18}}$	3479941204398616	`d218`
19	${s_0}^{e^{19}}$	810645248255668	`a6b4`

RSA PRG can have short cycles

RSA PRG with $N = 5154904286740261$ and $e = 3$.

Iteration	Value	State s_i	Output b_i
0	s_0	4047975530247052	`338c`
1	${s_0}^e$	2492861700191393	`34a1`
2	${s_0}^{e^2}$	4862773567328857	`9259`
...	...	...	...
16	${s_0}^{e^{16}}$	810645248255668	`a6b4`
17	${s_0}^{e^{17}}$	2887166220613321	`b6c9`
18	${s_0}^{e^{18}}$	3479941204398616	`d218`
19	${s_0}^{e^{19}}$	810645248255668	`a6b4`
20	${s_0}^{e^{20}}$	2887166220613321	`b6c9`
...	...	...	...

RSA PRG can have short cycles

- $s_i \equiv s_0^{e^i} \mod N$.
- We're in an exponent in an exponent
- Order of s_0 divides $\varphi(\varphi(N))$
- **Easy to generate parameters where period is very small factor of $\varphi(\varphi(N))$, giving short cycles**
- Such parameters are insecure... but cycling outputs would be visible to external user.

Observation 4

We can undetectably hide relations between RSA PRG states.

Candidate backdoor for RSA PRG: N embeds sparse relation

Simple relation gives obvious cycles:

$$\begin{aligned} e^i &\equiv e^j \mod \varphi(N) \\ \implies s_i &\equiv s_j \mod N \end{aligned}$$

Cycles (obvious)

But relation with more terms hides cycles:

$$\begin{aligned} e^h + e^i &\equiv e^j + e^\ell \mod \varphi(N) \\ \implies s_h \cdot s_i &\equiv s_j \cdot s_\ell \mod N \end{aligned}$$

No cycles, but still exploitable!

Candidate RSA PRG backdoor:
Choose N to encode a sparse relation between powers of e mod $\varphi(N)$. Exploit via multivariate Coppersmith method.

Example RSA PRG backdoor

Fix e. Choose a sparse relation like
$f(e) = e^{200} + e^{20} - e^{180} - e^{0} \equiv 0 \bmod \varphi(N)$.

Modulus generation:

1. Use ECM to find small factors p_i of $f(e)$.
2. Choose subsets S of factors and check if $1 + \prod_i p_i$ prime.
3. Repeat above until we have two factors.

Example RSA PRG backdoor

Fix e. Choose a sparse relation like
$f(e) = e^{200} + e^{20} - e^{180} - e^{0} \equiv 0 \bmod \varphi(N)$.

Exploiting backdoor: Recall $s_i \equiv s_0^{e^i} \mod N$.

$$e^{200} + e^{20} \equiv e^{180} + e^{0} \bmod \varphi(N)$$

$$s_0^{e^{200}} \cdot s_0^{e^{20}} \equiv s_0^{e^{180}} \cdot s_0^{e^{0}} \bmod N$$

$$s_{200} \cdot s_{20} \equiv s_{180} \cdot s_0 \bmod N$$

$$(2^k r_{200} + b_{200})(2^k r_{20} + b_{20}) \equiv (2^k r_{180} + b_{180})(2^k r_0 + b_0) \bmod N$$

A simple multivariate Coppersmith construction can solve for $|r_i| < N^{1/8}$.

Unclear how to get backdoor to work for Micali-Schnorr

Truncation prevents us from building exploitable relations

- RSA PRG has an elegant closed form: $s_i = {s_0}^{e^i}$
- MS does not: $s_i = ((((s_0{}^e - b_1)/2^k)^e - b_2)/2^k \ldots$

Trying to extend this idea results in a polynomial with exponentially many terms in the number of outputs.

Open problem: Need further ideas to extend candidate backdoor to Micali-Schnorr.

(e.g. Each state recurrence relation has few terms; can we somehow solve without expanding?)

Thoughts/Discussion

- **Open problem:** Is there a Gröbner basis approach? System is underconstrained without size limits on solutions. We tried various methods to constrain solution size but none worked.
- This problem has nerd-sniped generations of cryptographers, and after this talk hopefully it has nerd-sniped you too.
- We have never heard of anyone using Micali-Schnorr in the real world.
- Micali-Schnorr will be removed from ISO 18031 in next revision.