Zeroizing Attacks on Multilinear Maps

Tancrède Lepoint

Based on works of Brakerski, Coron, Cheon, Gentry, Halevi, Jia, Han, Hu, Lee, Maji, Miles, Raykova, Ryu, Sahai, Stehlé, Tibouchi, and discussions with many others

Inspired by Halevi's invited talk at CRYPTO 2015

Bochum - October 8, 2015 — Workshop on Tools for Asymmetric Cryptanalysis

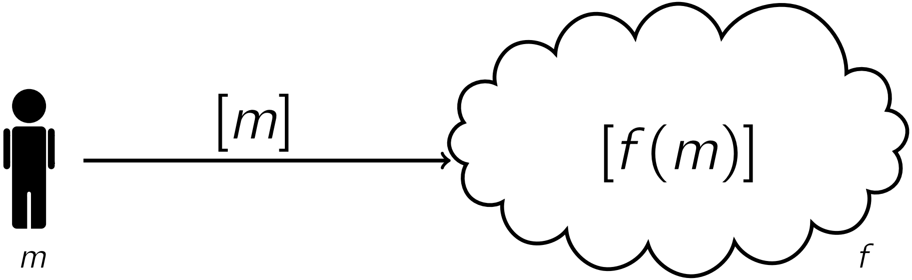
Outline

- Introduction & timeline
- Syntax of MMAPs
- The GGH13 Candidate
- “Zeroizing”, again and again
- Conclusion & open problems

Outline

- Introduction & timeline
- Syntax of MMAPs
- The GGH13 Candidate
- "Zeroizing", again and again
- Conclusion & open problems

compute on **hidden data** in a **non-interactive way**



data is **hidden** by encoding it

(for multilinear maps, a **test** will be possible on $[f(m)]$)

CryptoExperts

example: **discrete logarithm**

- m is encoded as $[m] = g^m$ (in some group G)
  - Recovering m from $[m]$ is hard (discrete log)
- Compute linear functions is **easy**
  - $\prod_i [m_i]^{u_i} = \left[\sum_i u_i m_i\right]$
  - Can check whether $m = 0$
- Computing other functions **seems hard**
  - $[m_1], [m_2] \mapsto [m_1 \cdot m_2]$ (Diffie-Hellman)
  - Even testing an alleged solution is hard $[m_1 \cdot m_2] \approx_c u$ (Decisional DH)

"DDH assumption is a gold mine" [Boneh98]

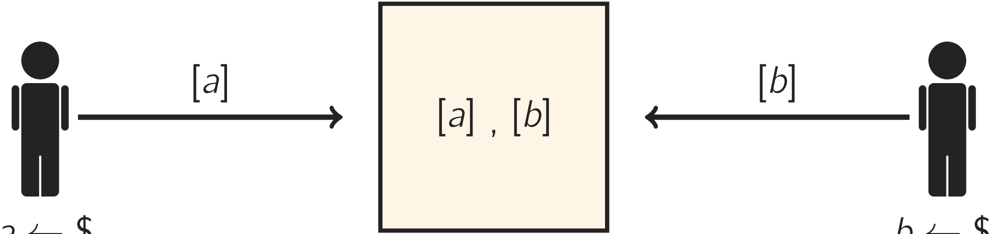
DLog cryptography has **many applications**

(e.g. CCA-secure PKE, commmitments, zero-knowledge proofs, etc.)

“DDH assumption is a gold mine” [Boneh98]

DLog cryptography has **many applications**

(e.g. CCA-secure PKE, commmitments, zero-knowledge proofs, etc.)



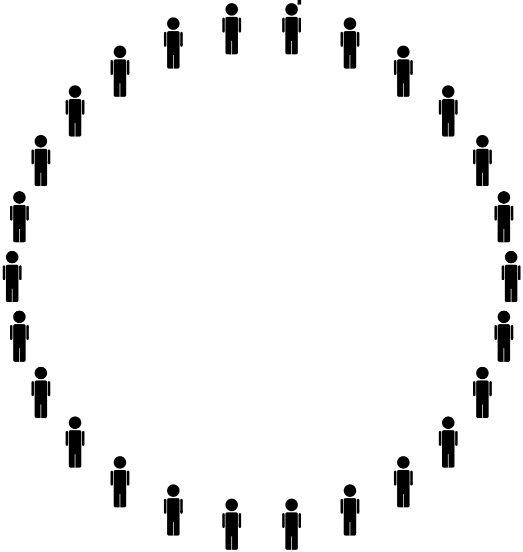
$a \leftarrow \$$ $\qquad\qquad$ $b \leftarrow \$$

$K = [b]^a = g^{ab}$ $\qquad\qquad$ $K = [a]^b = g^{ab}$

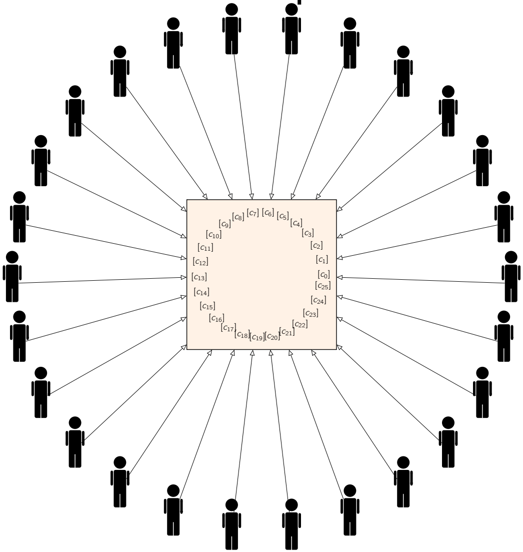
beyond DDH: **bilinear maps**

- m is encoded as $[m]_1 = g^m$ (in group G_1)
- map $e([m_1]_1^a, [m_2]_1^b) = [m_1 \cdot m_2]_2^{ab}$ (in group G_2)
- in bilinear-map group, computing quadratic functions in the exponent is **easy**
  - but computing/checking cubics **seems hard**
- Many new applications
  - 3-partite DH Key Exchange
  - Efficient NIZK proofs
  - ABE/functional encryption for simple func.
  - Broadcast Encryption, Traitor Tracing, ...

can we go beyond **2**-linear maps?

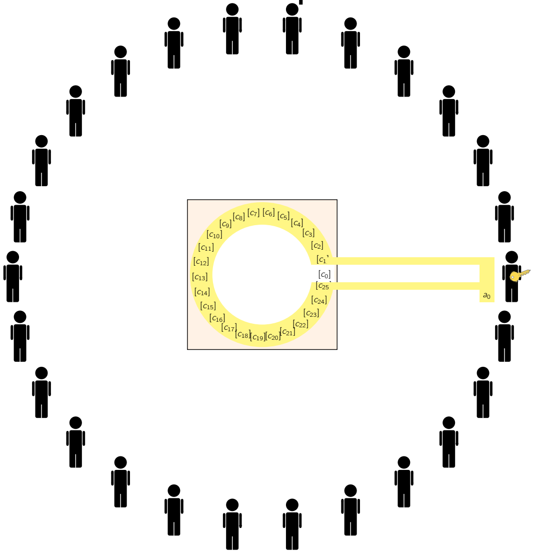


can we go beyond **2**-linear maps?

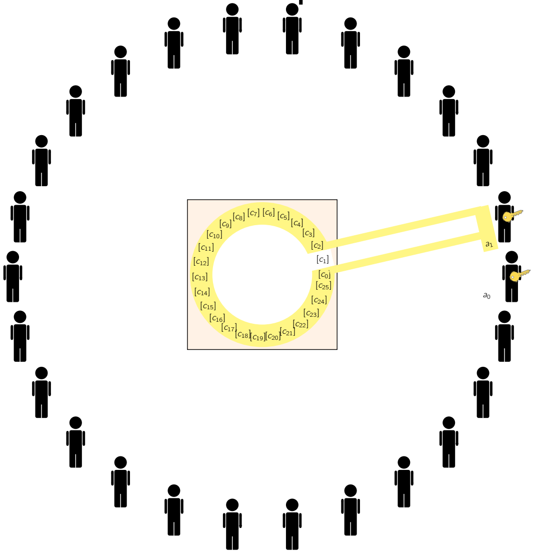


CryptoExperts

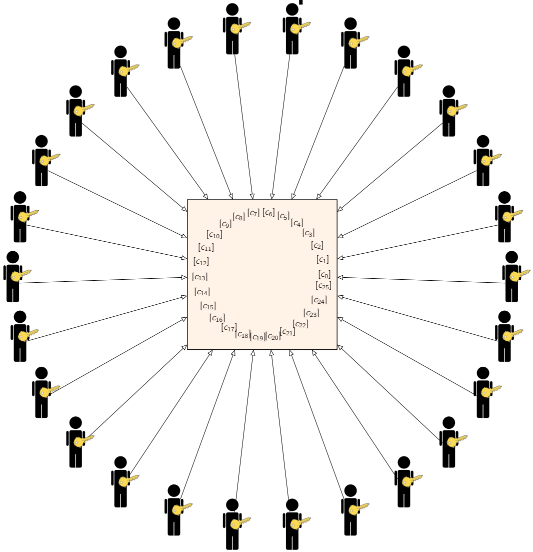
can we go beyond **2**-linear maps?



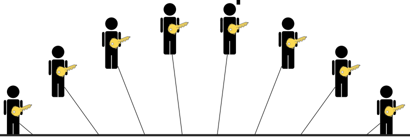
can we go beyond **2**-linear maps?



can we go beyond **2**-linear maps?



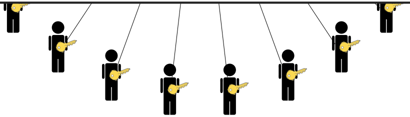
can we go beyond **2**-linear maps?



It would be useful... [BS03]

. . . but **seems hard to get**

from the realm of algebraic geometry



MMAPs are similar to Somewhat HE

MMAPs

- ✓ Encoding m into $[m] = g^m$
- ✓ Computing low-degree polynomials of the $[m]$'s is easy
- ✓ Can test for zero but cannot recover m

SWHE

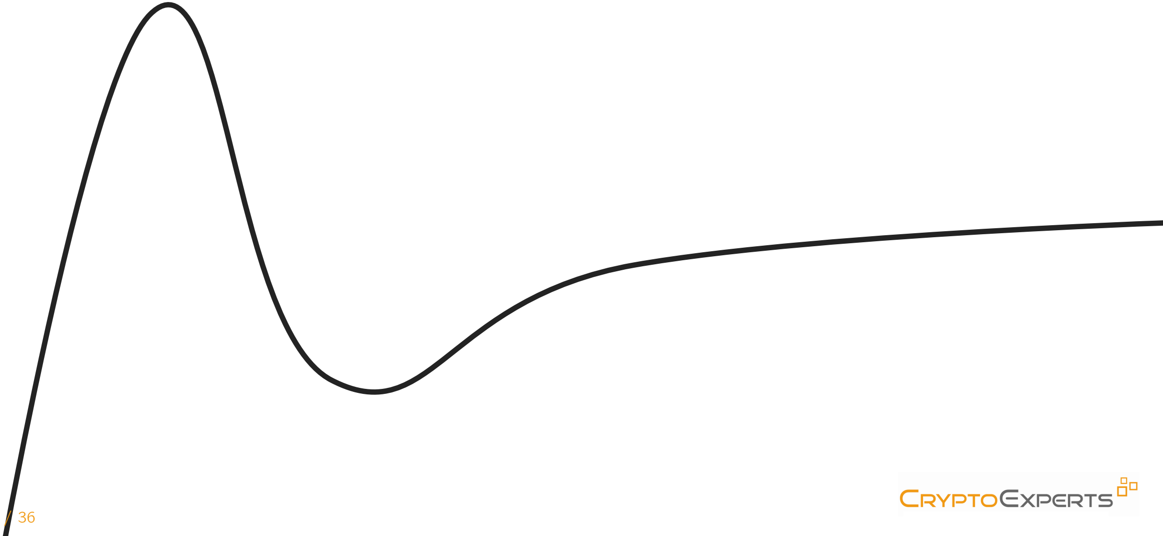
- ✓ Encrypting m into $c_m = E(m)$
- ✓ Computing low-degree polynomials of the c_m's is easy
- × Cannot test anything (except with the secret key)

CryptoExperts

main ingredient: **testing for zero**

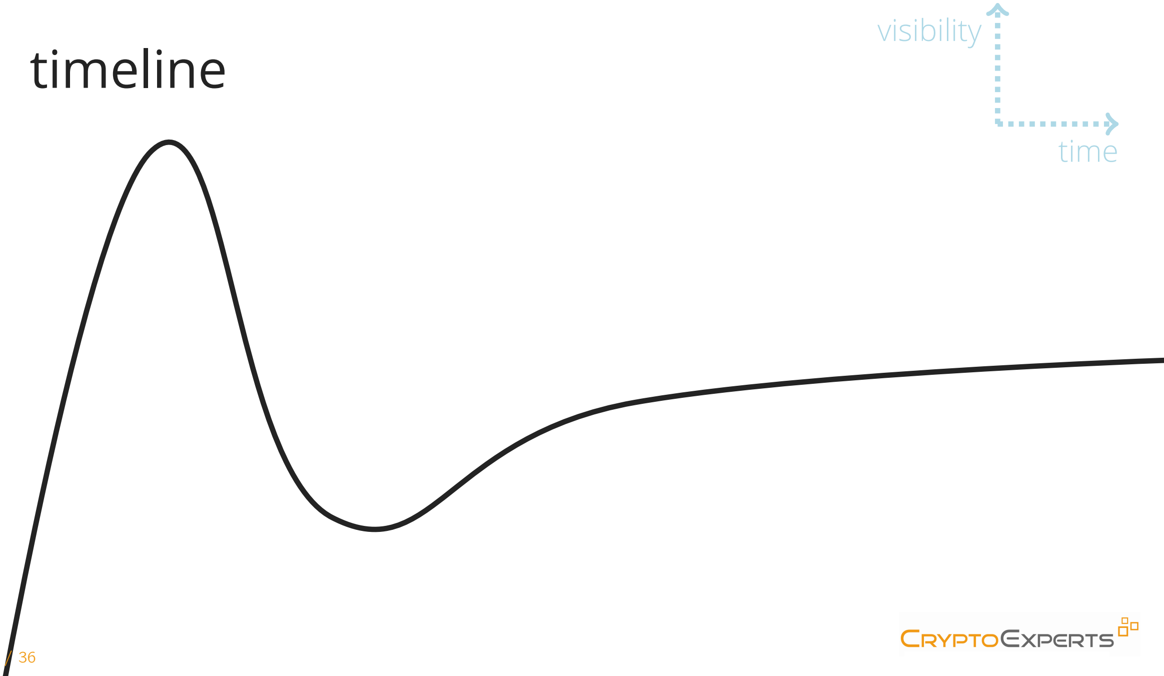
- To be useful, MMAPs should have the ability to test whether two degree-κ expressions are equal
  - Same as testing whether a degree-κ expr. is 0
- Current solutions: take a SWHE scheme and publish an “handicapped” version of the SK
  - called **zero-test parameter**
  - can identify enc. of 0, but cannot decrypt (large plaintext space)

timeline: the **hype cycle** of MMAPs

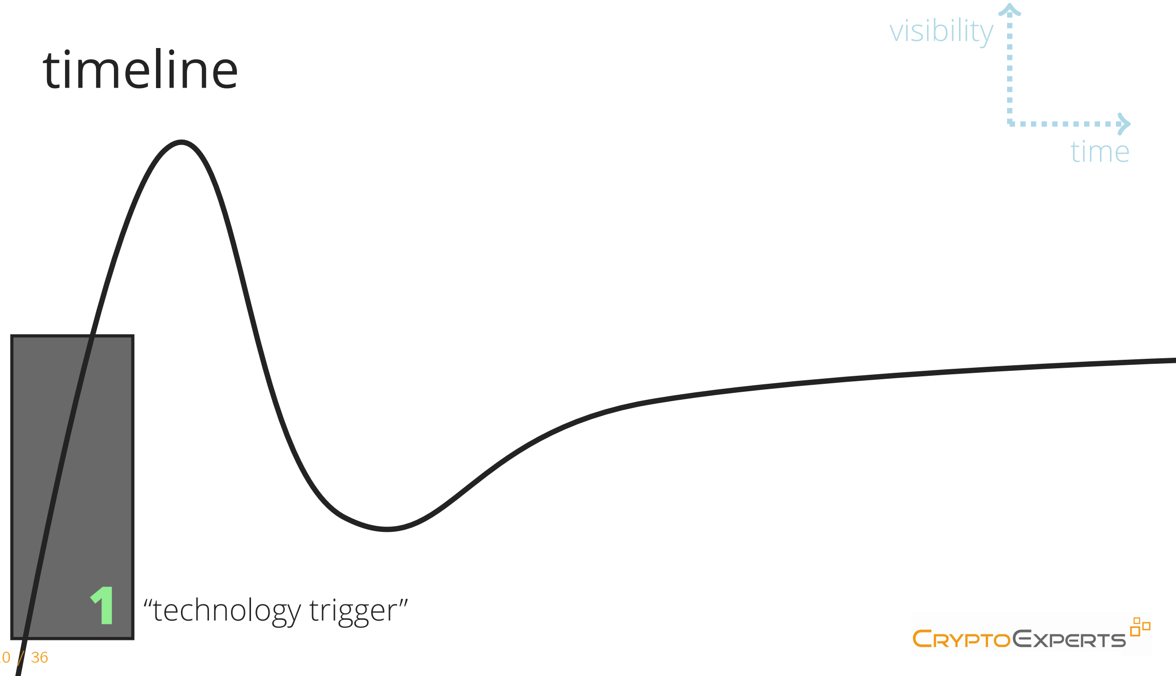


CryptoExperts

timeline

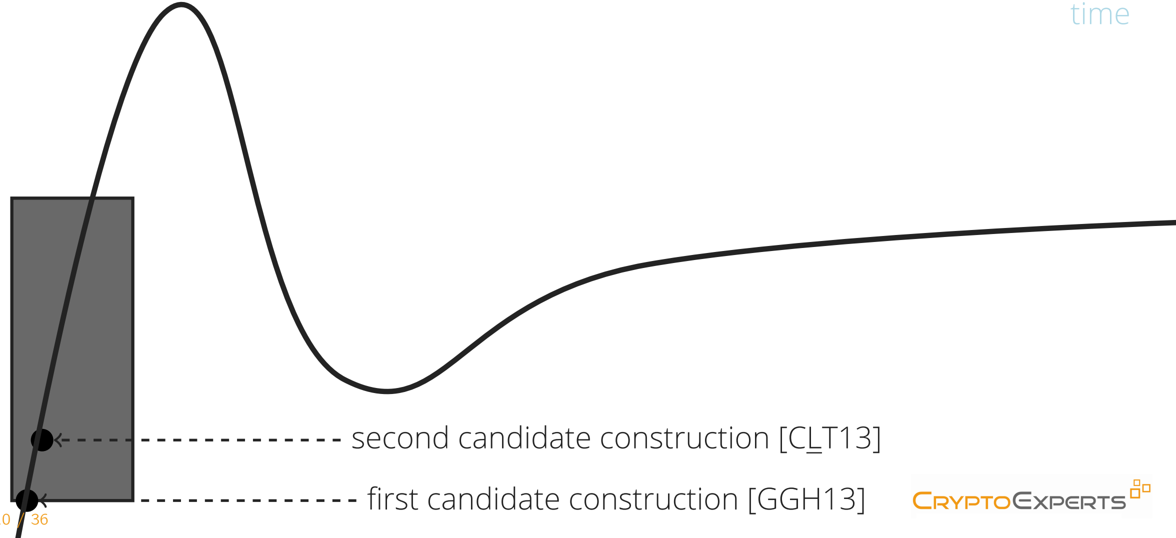


timeline

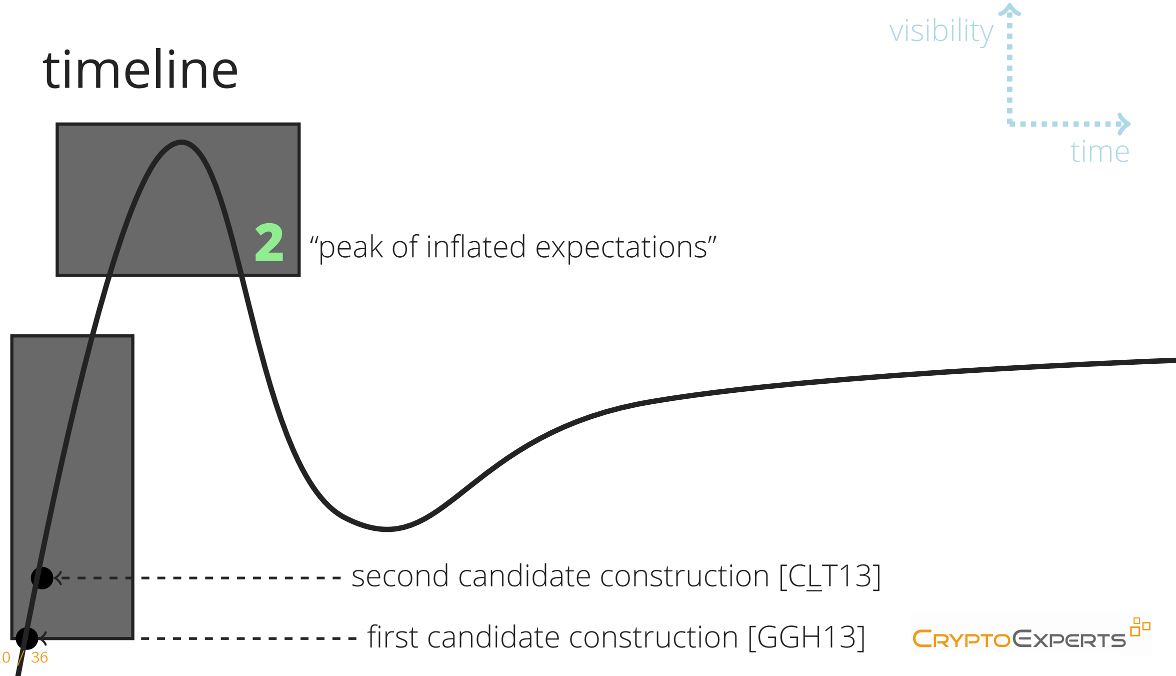


timeline

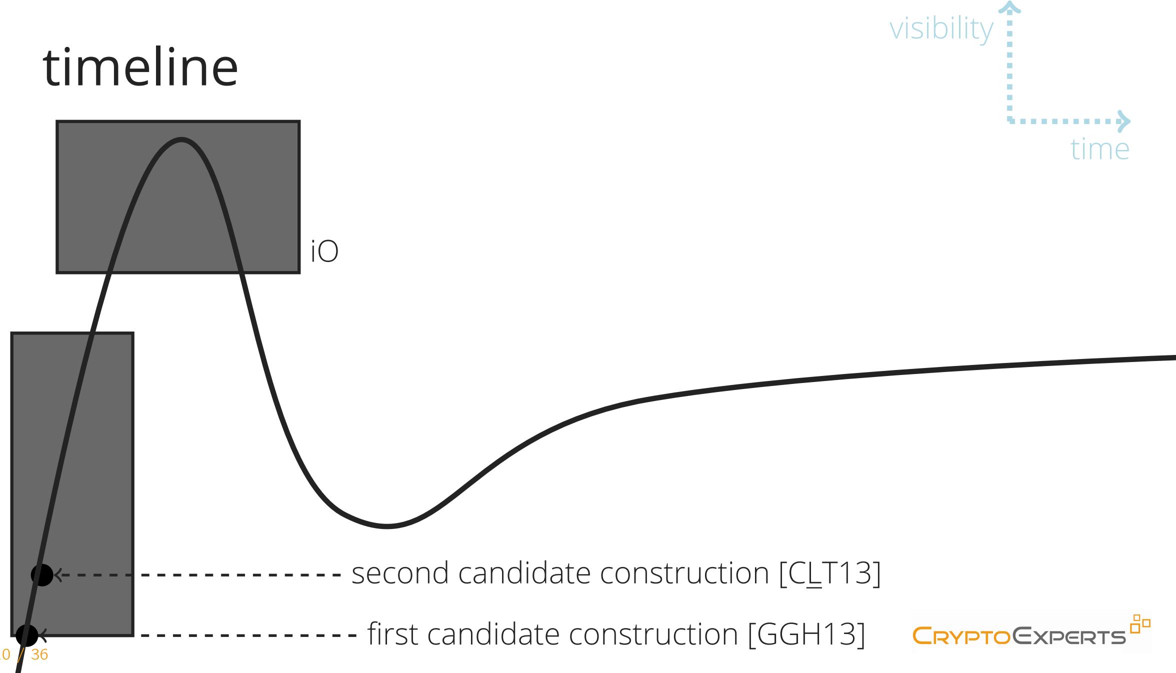
visibility
time


second candidate construction [CLT13]
first candidate construction [GGH13]

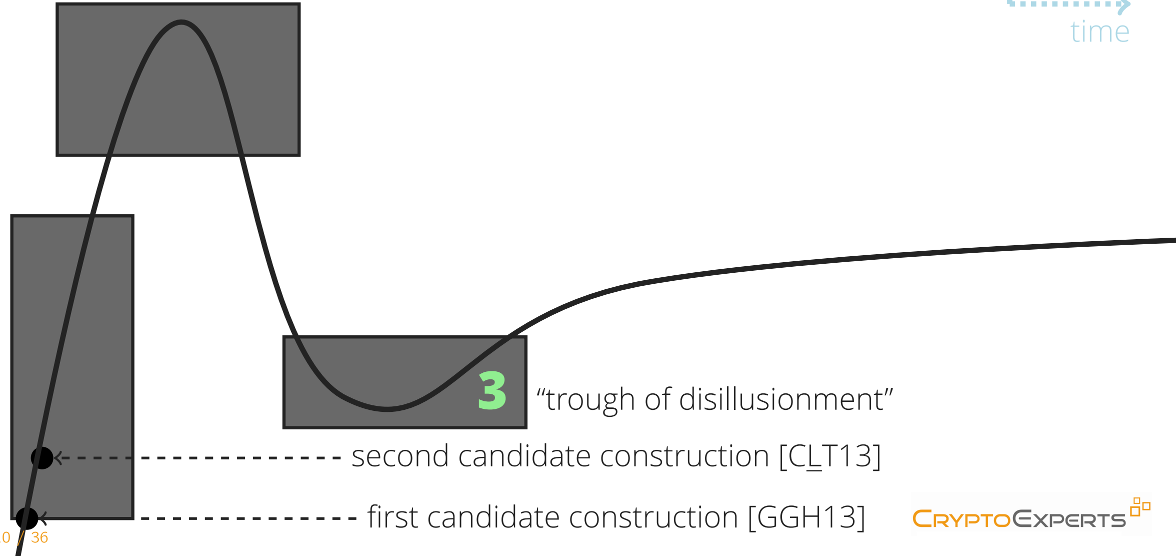
timeline


visibility
time
2
“peak of inflated expectations”
second candidate construction [CLT13]
first candidate construction [GGH13]

timeline

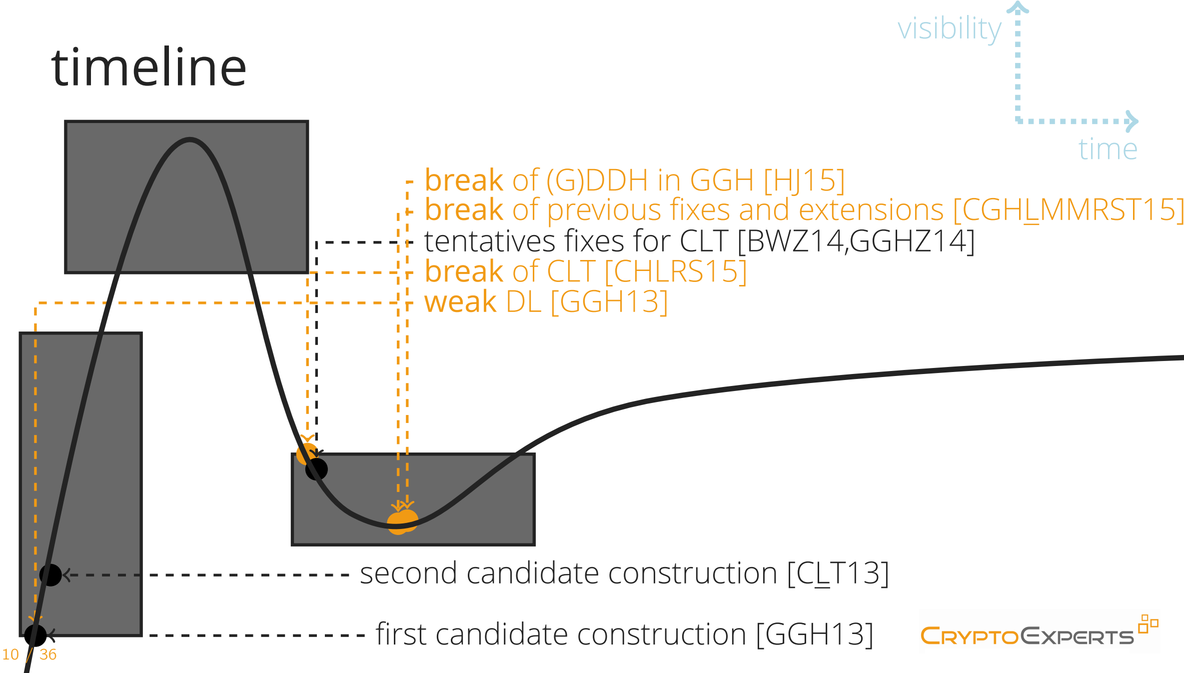

visibility
time
iO
second candidate construction [CLT13]
first candidate construction [GGH13]

timeline



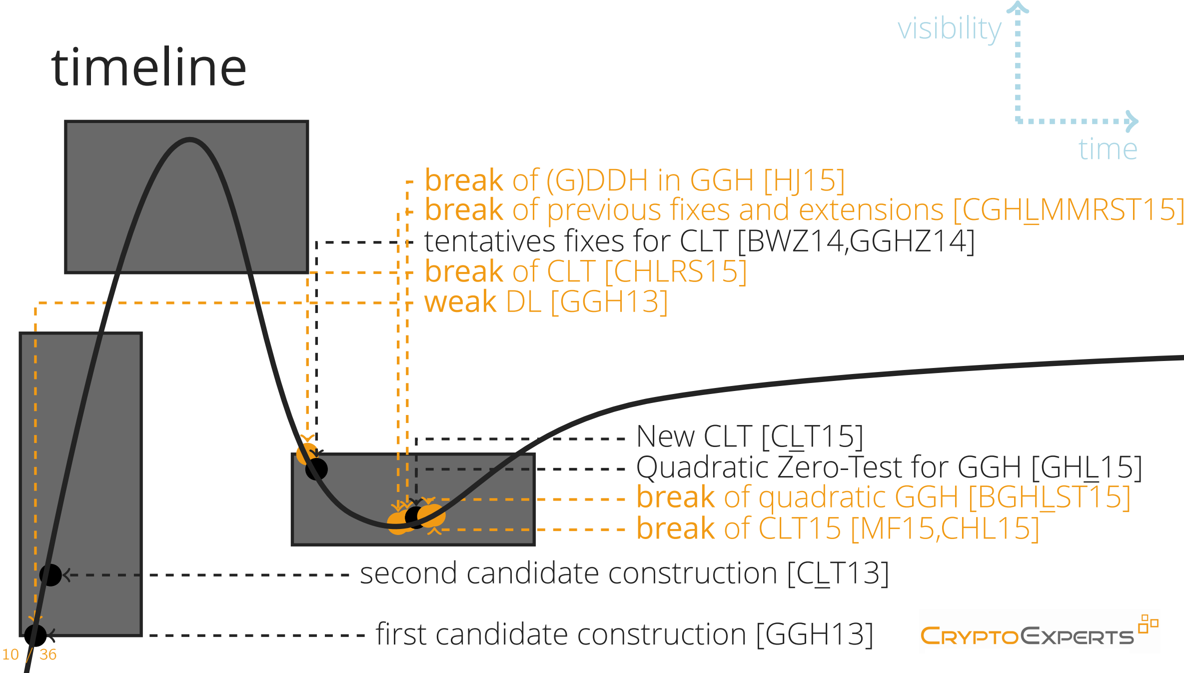
CryptoExperts

timeline



CryptoExperts

timeline



CryptoExperts

Outline

- Introduction & timeline
- Syntax of MMAPs
- The GGH13 Candidate
- “Zeroizing”, again and again
- Conclusion & open problems

syntax of MMAPs

- All constructions expose somewhat different interfaces.
- Syntax proposed by [Hal15] in three parts
  - **Initialization**: generation of public/secret parameters
    - Also define “plaintext space” and “encoding space”
  - **Encoding**: use the **secret parameters** to encode plaintexts
  - **Operations**: use the **public parameters** to add, multiply and test for 0
    - (with restrictions)

restricting operations with **tags**

- Each encoding has a **tag**
- **Add** elements with the same tag
- **Multiply** elements with compatible tags
  - Resulting tag follow simple rule
- **Zero-Test** only an encoding at a **distinguished tag** (top-level)

CryptoExperts

restricting operations with **tags**

- Each encoding has a **tag**
- **Add** elements with the same tag
- **Multiply** elements with compatible tags
  - Resulting tag follow simple rule
- **Zero-Test** only an encoding at a **distinguished tag** (top-level)

Examples:

- $\mathcal{T} = \{1, 2, \ldots, \kappa\}$, addition of tags during multiplication, test at level κ
- DAG [GGH15,Hal15]

security of MMAPs

- DL security: hard to recover m from $[m]_i$
- hard to distinguish zeros at tags $i \neq \kappa$ (except by lifting them up)
- generalized DDH: hard to identify relations for incompatible tags
- etc.

CryptoExperts

security of MMAPs

- DL security: hard to recover m from $[m]_i$
- hard to distinguish zeros at tags $i \neq \kappa$ (except by lifting them up)
- generalized DDH: hard to identify relations for incompatible tags
- etc.

- *Attacks on MMAPs often do not apply to obfuscation because everything is* **glued** *there: only "allowed operations" can be performed meaningfully*

Outline

- Introduction & timeline
- Syntax of MMAPs
- The GGH13 Candidate
- "Zeroizing", again and again
- Conclusion & open problems

GGH13 candidate

- Works in cyclotomic rings $R = \mathbb{Z}[x]/\Phi_m(x)$
  - work modulo a large integer $q \approx 2^{\sqrt{m}}$
  - define $R_q = R/qR$
- Secrets parameters:

$$z \leftarrow R_q, \qquad \text{“small”}\ g \in R$$

- Plaintext space: $R_g = R/gR$
  - g is chosen so that $R_g \simeq \mathbb{F}_p$ for some prime p
  - but g, p are not made public
- Level-i encoding of $m \in R_g$:

$$u = [(m + r \cdot g)/z^i]_q$$

CryptoExperts

zero testing parameter

- Level-κ encoding of 0:

$$u = [r \cdot g / z^{\kappa}]_q$$

- Zero-Test parameter (*h* small-ish):

$$p_{zt} = [h \cdot z^{\kappa} / g]_q$$

- Multiplying we get $|[u \cdot p_{zt}]_q| = |r \cdot h| \ll q$
- If u doesn't encode 0, we get $|[u \cdot p_{zt}]_q| \approx q$

GGH13 properties

- Encoding is related to a numerator $u \sim (e)$ $(e = m + r \cdot g)$
  - Finding e means breaking the scheme
  - An encoding of 0 is $u \sim (rg)$
- Adding / multiplying encodings operate on the numerators over R (not modulo q)

$$u_1 + u_2 \sim (e_1 + e_2), \qquad u_1 \cdot u_2 \sim (e_1 \cdot e_2)$$

- Zero-testing top-level encodings $u \sim (rg)$ we get $ztst(u) = r \cdot h$ over R (no mod q)

CryptoExperts

Outline

- Introduction & timeline
- Syntax of MMAPs
- The GGH13 Candidate
- “Zeroizing”, again and again
- Conclusion & open problems

zeroizing attack against GGH13

- First "zeroizing attack", known from the beginning [GGH13]

Setting:

- A level-k encoding of 0: $u_0 \sim (r_0 g)$
- Many level-$\kappa - k$ encodings: $u_j \sim (e_j)$

Zeroizing:

- Compute $u_0 u_j \sim (e_j r_0 g)$ and zero-test it

$$y_j = ztst(u_0 u_j) = hr_0 \cdot e_j$$

- We recovered the e_j's up to a fector $h' = hr_0$
  - Find and remove h' by computing GCD's in R?

zeroizing attack against GGH13

- GCDs give the ideals $e_j R$ and not the e_j themselves
  - Moreover, $e_j R$ carries no info on $e_j + gR$ (if e_j and g are coprime)
  - but if we have many $e_j \in gR$, we can recover gR
- Knowing gR and $e'_j = \{h' \cdot e_j\}$ is enough to break many assumptions (e.g. SubM, DLIN)

zeroizing attack against GGH13

- GCDs give the ideals $e_j R$ and not the e_j themselves
  - Moreover, $e_j R$ carries no info on $e_j + gR$ (if e_j and g are coprime)
  - but if we have many $e_j \in gR$, we can recover gR
- Knowing gR and $e_j' = \{h' \cdot e_j\}$ is enough to break many assumptions (e.g. SubM, DLIN)

Encodings of 0×0 are harmful because they let you recover gR

CryptoExperts

zeroizing attack against GGH13

- GCDs give the ideals $e_j R$ and not the e_j themselves
  - Moreover, $e_j R$ carries no info on $e_j + gR$ (if e_j and g are coprime)
  - but if we have many $e_j \in gR$, we can recover gR
- Knowing gR and $e_j' = \{h' \cdot e_j\}$ is enough to break many assumptions (e.g. SubM, DLIN)

Encodings of 0×0 are harmful because they let you recover gR

For current MMAPs, what is important is to know *which distributions* are safe to encode

CryptoExperts

attempted fix #1: matrix GGH [GGHZ14]

- Encoding = matrix, plaintext = eigenvalue
- Secret parameters: z, g and random $P \in R_q^{n\times n}$, small vectors $\vec{s}, \vec{t} \in R^n$
- To encode $\alpha \in R_g$, choose small $E \in R^{n\times n}$ s.t.

$$\vec{s} \times E = \alpha \cdot \vec{s} \bmod gR$$

  - The encoding is $U = [P \cdot E \cdot P^{-1}/z^i]_q$

CryptoExperts

matrix GGH zero testing

- Top-level encoding of 0 is s.t.

$$U = [P \cdot E \cdot P^{-1}/z^{\kappa}]_q \quad \text{and} \quad \vec{s} \cdot E = g \cdot \vec{r}$$

- Zero-test parameter: $(\vec{s'}, \vec{t'})$ where $\vec{s'} = [z^{\kappa}/g \cdot \vec{s} \cdot P^{-1}]_q$, $\vec{t'} = [P \cdot \vec{t}]_q$

- Multiplying, we get

$$|[\vec{s'} \cdot U \cdot \vec{t'}]_q| = |\langle r, \vec{t} \rangle| \ll q$$

CryptoExperts

zeroizing attack against matrix GGH [CGHLMMRST15]

- Very similar to Cheon et al. attack on CLT13 [CHLRS15]

Setting:

- Many level-ℓ encodings of 0: $u_i \sim (A_i)$ s.t. $\vec{s} \cdot A_i = g \cdot \vec{a}_i$
- Some level-ℓ' encodings of 0: $v_j \sim (B_j)$, s.t. $\vec{s} \cdot B_j = g \cdot \vec{b}_j$
- Many level-$\kappa - \ell - \ell'$ encodings: $w_k \sim (C_k)$

Zeroizing:

- Compute $u_i v_j w_k$ and zero-test it

$$y_{ijk} = ztst(u_i v_j w_k) = (\vec{s} A_i / g) \cdot B_j \cdot (C_k \cdot \vec{t})$$

- Construct a matrix over R by varying i, k:

$$Y_j = \tilde{A} \cdot B_j \cdot \tilde{C}$$

zeroizing attack against matrix GGH [CGHLMMRST15]

Computing GCD's:

- $Y_j = \tilde{A} \cdot B_j \cdot \tilde{C}$, therefore

$$\det(Y_j) = \det(\tilde{A}) \cdot \det(B_j) \cdot \det(\tilde{C})$$

- whp $\gcd(\det(Y_1), \det(Y_2), \ldots) = \det(\tilde{A}) \cdot \det(\tilde{C})$
- We get $\det(B_j) \cdot R$ for all j

Encodings of 0×0**:**

- Recall that $\vec{s} \cdot B_j = 0 \bmod gR$, so $\det(B_j)$ is divisible by g
- With some luck, $gR = \gcd(\det(B_1), \det(B_2), \ldots)$
- Same weakness as before: the fix failed

CryptoExperts

attempted fix #2: quadratic GGH [GHL15]

- **Moral so far:**
  - Recovering gR allows to break assumptions: *primary goal*
  - Attacks rely on the **linear** form of zero-testing

CryptoExperts

attempted fix #2: quadratic GGH [GHL15]

- **Moral so far:**
  - Recovering gR allows to break assumptions: *primary goal*
  - Attacks rely on the **linear** form of zero-testing
- **Let's try to make the zero-testing non-linear!**

attempted fix #2: quadratic GGH [GHL15]

- **Moral so far:**
  - Recovering gR allows to break assumptions: *primary goal*
  - Attacks rely on the **linear** form of zero-testing
- **Let's try to make the zero-testing non-linear!**
- Every coefficient of $y = [p_{zt} \cdot u]_q$ is $\mathbb{Z}_q$-linear in the coeff. of u
  - We have $\phi(m)$ such linear functions $\ell_i(u)$
- Consider a quadratic (or more) function

$$z(u) = \sum_{i,j} \alpha_{ij} \cdot \ell_i(u) \cdot \ell_j(u)$$

α_{ij} smallish s.t. $|z(u)| \ll q$ when u encodes 0 and otherwise $\approx q$

zeroizing attack against quadratic GGH [BGHLST15]

- Key idea: compute the derivative to get back to a linear zero-testing
- Derivative of $p(x_1, \ldots, x_n)$ in $\vec{a}$:

  $$p'_{\vec{a}}(x_1, \ldots, x_n) = p(x_1 + a_1, \ldots, x_n + a_n) - p(x_1, \ldots, x_n) \bmod q$$

zeroizing attack against quadratic GGH [BGHLST15]

- Key idea: compute the derivative to get back to a linear zero-testing
- Derivative of $p(x_1, \ldots, x_n)$ in $\vec{a}$:

$$p'_{\vec{a}}(x_1, \ldots, x_n) = p(x_1 + a_1, \ldots, x_n + a_n) - p(x_1, \ldots, x_n) \bmod q$$

Setting:

- Two top levels encodings of 0 u and v

Derivation:

- Compute the derivative of z in u and apply it on v:

$$z'_u(v) = z(u + v) - z(v) = \sum \rho_i v_i + \rho'$$

- $\rho' = z'_u(0) = z(u) - z(0) = z(u) \ll q$

zeroizing attack against quadratic GGH [BGHLST15]

- We deduce

$$|z'_u(v)| = |\sum \rho_i v_i| \ll q$$

zeroizing attack against quadratic GGH [BGHLST15]

- We deduce

$$|z'_u(v)| = |\sum \rho_i v_i| \ll q$$

Using the structure of R (assume $R = \mathbb{Z}[x]/(x^n+1)$ for simplicity):

- Define $r = \rho_0 - \sum \rho_{n-i} x^i$
- We have

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} \rho_0 & \rho_1 & \cdots & \rho_{n-1} \\ -\rho_{n-1} & \rho_0 & \cdots & \rho_{n-2} \\ \vdots & \vdots & \vdots & \vdots \\ -\rho_1 & -\rho_2 & \cdots & \rho_0 \end{pmatrix} \cdot \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{pmatrix}$$

- Every $y_i = p'_u(-x^{n-i}v) - \rho'$, thus small

attempted fix #3: graph GGH13 [Hal15,Cor15]

- Halevi suggested to use DAG for the tags in GGH13 (idem as in GGH15)
  - Encoding of an element $\alpha \in R_g$ is

$$\tilde{C} = P^{-1} \cdot C \cdot P$$

where C is the matrix “multiply by small $c \in \alpha + gR$”

CryptoExperts

attempted fix #3: graph GGH13 [Hal15,Cor15]

- Halevi suggested to use DAG for the tags in GGH13 (idem as in GGH15)
  - Encoding of an element $\alpha \in R_g$ is

$$\tilde{C} = P^{-1} \cdot C \cdot P$$

    where C is the matrix “multiply by small $c \in \alpha + gR$”

- Unfortunately this fix does not hold either [Cor15]
  - extension of [CHLRS15,CGHLMMRST15] using the Cayley Hamilton theorem

Outline

- Introduction & timeline
- Syntax of MMAPs
- The GGH13 Candidate
- “Zeroizing”, again and again
- Conclusion & open problems

conclusion: security landscape

- Zeroizing attacks are **devastating** for multilinear maps [GGH13,CLT13]
  - Break many assumptions and schemes
  - But not all (e.g. obfuscation is mainly unaffected!)
- All attempts made public at strengthening these schemes are broken!
  - the attempts to make "zero-testing" less linear failed [CLT15,GHL15]
- Similar situation for [GGH15]
- Break & Repair mode
  - LOTS of room for more cryptanalysis and more theory

CryptoExperts

state-of-the-art of today afaik

- GGH13: weak distributions 0×0
  - all fixes broken [GGHZ14,GHL15,Hal15,≥ 5 unpublished attempts I know about]
- CLT13: “too many” encodings of 0
  - early fixes broken [GGHZ14,BWZ14]
  - new CLT [CLT15] **completely broken** by [MF15,CHL15] (last week on Eprint): thus weaker than CLT13
- GGH15: some “low-level” encodings of 0
- Gu’s MMAP [Gu15]: completely broken [PS15]

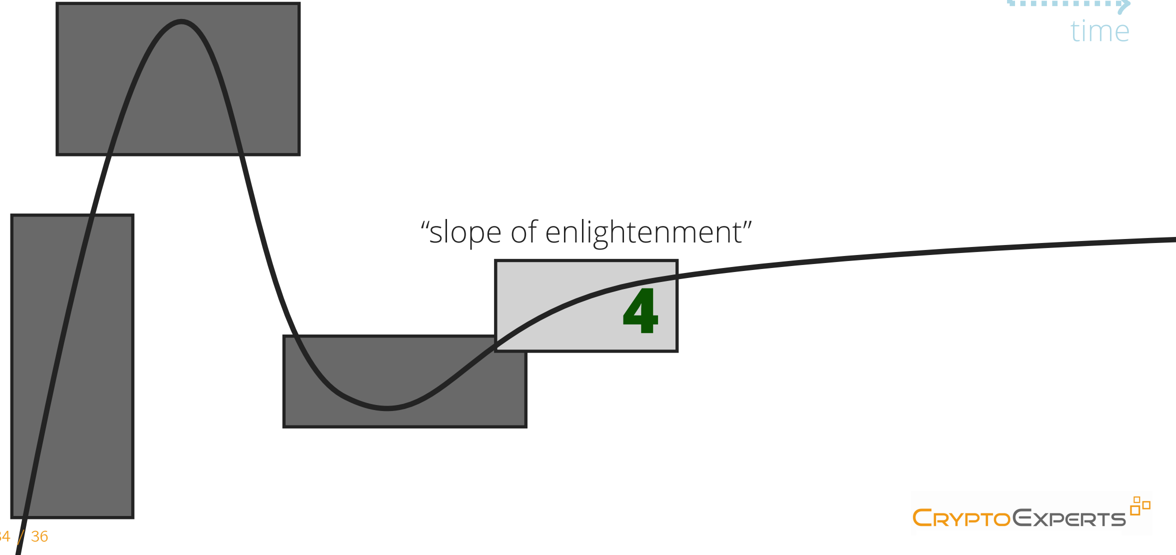
CryptoExperts

open problems

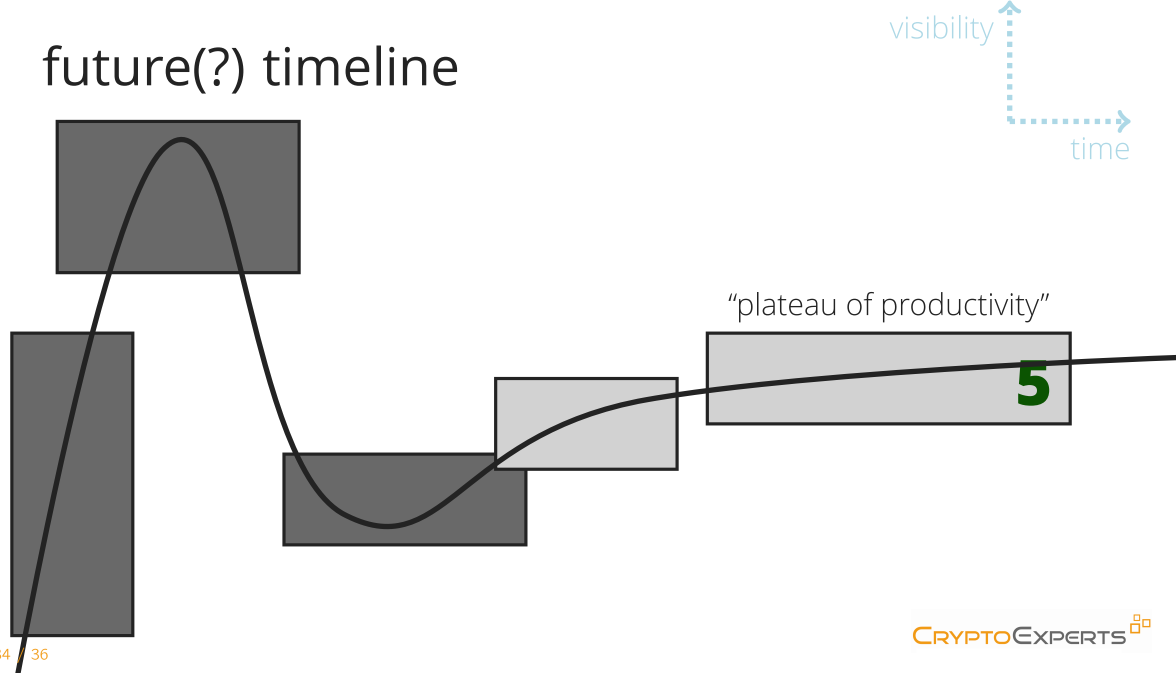
open problems

Everything.

future(?) timeline

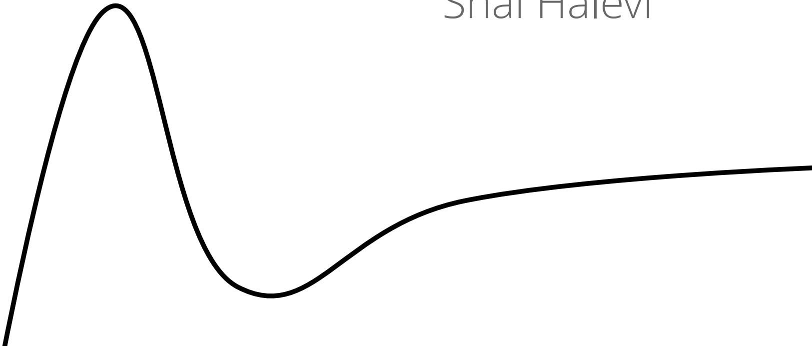


future(?) timeline



“This is going to be a bumpy ride”

Shai Halevi



Questions?

https://www.cryptoexperts.com/tlepoint