Affine Determinant Programs

Mark Zhandry (NTT Research & Stanford University)

Based on joint work with James Bartusek, Yuval Ishai, Aayush Jain, Fermi Ma, and Amit Sahai

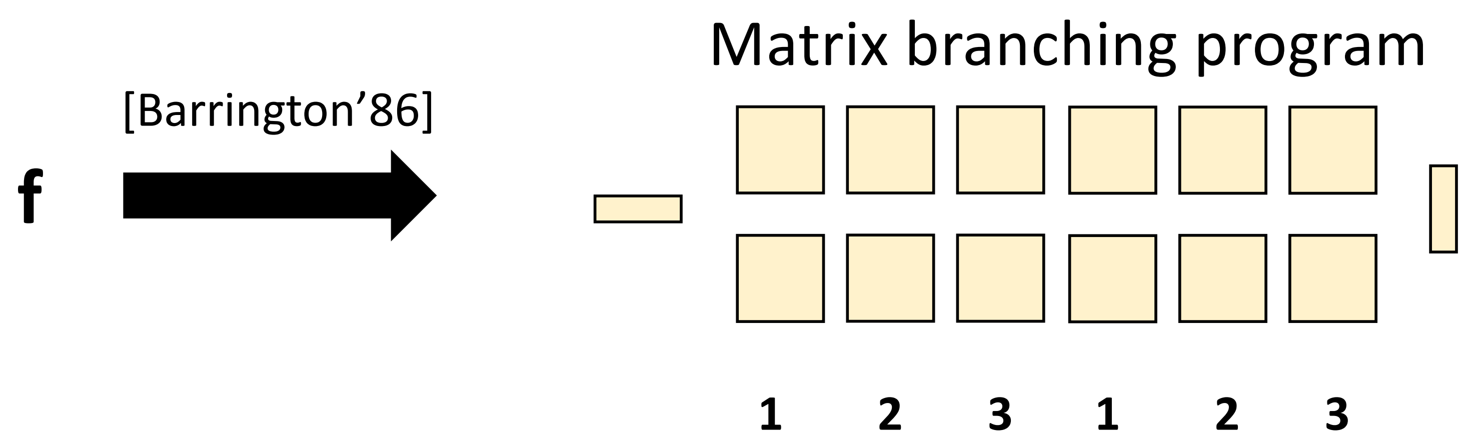
Motivation: different structures for obfuscating programs

Potential advantages:

- Improved efficiency
- Different hardness assumptions

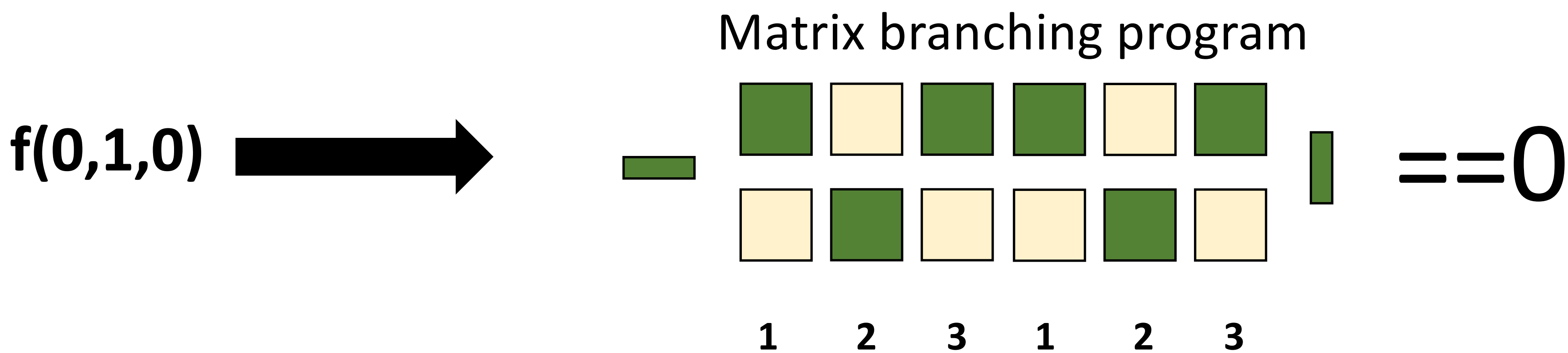
The original obfuscation approach
[Garg-Gentry-Halevi-Raykova-Sahai-Waters’13]

Suffices to consider functions **f** in **NC¹**



The original obfuscation approach
[Garg-Gentry-Halevi-Raykova-Sahai-Waters’13]

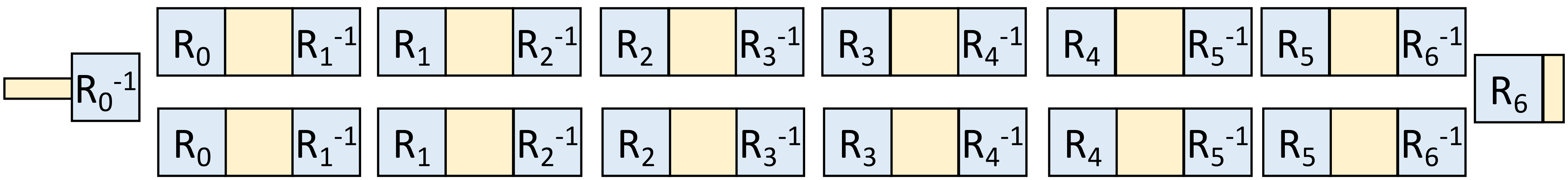
Suffices to consider functions **f** in **NC^1**



The original obfuscation approach

[Garg-Gentry-Halevi-Raykova-Sahai-Waters’13]

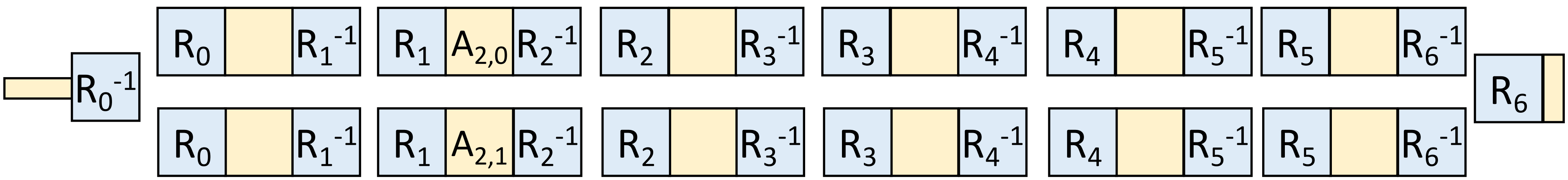
[Kilian’88]



Thm [Kilian’88]: For any 1 input, matrix distribution only depends on output, independent of circuit

The original obfuscation approach

[Garg-Gentry-Halevi-Raykova-Sahai-Waters’13]



Problem: still lots of information revealed

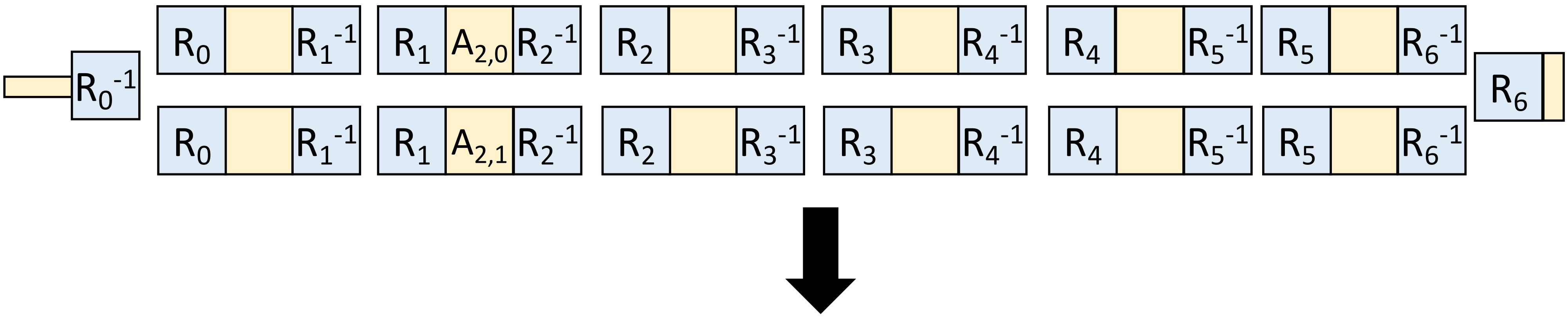
Mix-and-match attacks: read different input bits each time

$$R_1 A_{2,0} R_2^{-1} \left(R_1 A_{2,1} R_2^{-1} \right)^{-1} = R_1 A_{2,0} A_{2,1}^{-1} R_1^{-1}$$

Same eigenvalues as $A_{2,0} A_{2,1}^{-1}$

The original obfuscation approach

[Garg-Gentry-Halevi-Raykova-Sahai-Waters'13]



Encode in "multilinear map"

Huge efficiency losses, plus security of known multilinear maps questionable

Q: Can we get security “directly” without encoding?

What is an affine-determinant program?

Program $\mathbf{f : \{0,1\}^n \rightarrow \{0,1\}}$ described by $\mathbf{n+1}$ matrices:

$$\boxed{A_0} \quad \boxed{A_1} \quad \cdots \quad \boxed{A_n}$$

$$\mathbf{f(x):}\ \ \mathbf{Det}\left(\boxed{A_0} + x_1 \times \boxed{A_1} + \ldots + x_n \times \boxed{A_n} \right) == 0$$

Arithmetic over some finite commutative ring, e.g. $\mathbf{Z_p}$

Used previously in [Ishai-Kushilevitz'02,…] for somewhat unrelated goals

What is an affine-determinant program?

Program $\mathbf{f : \{0,1\}^n \rightarrow \{0,1\}}$ described by matrix of affine funcs:

$$\boxed{\mathbf{A(X)}}$$

$$\mathbf{f(x):\ Det\left(\boxed{A(x)}\right) == 0}$$

What functions can be computed by ADPs?

Thm [Ishai-Kushilevitz'02]: branching program → ADP

[IK] Re-randomization

$\mathbf{A(X)} \rightarrow \mathbf{B(X)} = \mathbf{R}\ \mathbf{A(X)}\ \mathbf{S}$ for random invertible **R,S**

Thm [Ishai-Kushilevitz'02]: For any input **x, B(x)** depends on **f(x)** but independent of **f**

Good: seems to prevent mix-and-match attacks!

Bad: $\mathbf{B(x).B(x')^{-1}}$ has same eigenvalues as $\mathbf{A(x).A(x')^{-1}}$

Even better re-randomization?

$$\boxed{A(X)} \rightarrow \boxed{B(X)} = \boxed{R}\left(\boxed{A(X)} + 2\boxed{E(X)}\right)\boxed{S}$$

Where:

- **Det(A(x)) ∈ {0,1}**
- **Det(R) = Det(S) = 1**
- Large field so that **Det(A(x)+2E(x))** doesn't wrap around

➡ **Det(B(x)) = Det(A(x)) mod 2 = f(x)**

An Attack on Even Noise

$$\mathbf{A(X)} \rightarrow \mathbf{B(X)} = \mathbf{R}\left(\mathbf{A(X)} + 2\,\mathbf{E(X)}\right)\mathbf{S}$$

If have guess for **A(X)**, can recover **E(X)**

Idea: **Det(B(X)) mod 4 = Det(A(X)+2E(X)) mod 4** is linear in **E(X)** since higher degree terms eliminated
→ Eval on many **x** to get many equations, and solve

Also give a more combinatorial randomization ("Random local substitutions") to hide even **A(X)**

Unclear how much security this adds, and has been attacked as well [Yao-Chen-Yu'21]

A direct construction

“Direct” candidate ADP obfuscator for $\mathbf{NC^1}$

Step 0: push all **NOT** gates to input wires

“Direct” candidate ADP obfuscator for $\mathbf{NC}^1$

Step 1: Obfuscation for input wires

$$\mathbf{f(X) = X_i :} \quad \boxed{\mathbf{A(X)}} \quad \mathbf{= u\,(X_i - 1)}$$

$$\mathbf{f(X) = 1\text{-}X_i :} \quad \boxed{\mathbf{A(X)}} \quad \mathbf{= u\,X_i}$$

“Direct” candidate ADP obfuscator for **NC^1**

Step 2: Recursively obfuscate sub-circuits

2a: $\mathbf{f(X) = g(X) \wedge h(X)}$:

$$A^f(X) = \underset{(2r-1)\times(2r)}{R} \begin{bmatrix} A^g(X) & 0 \\ 0 & A^h(X) \end{bmatrix} \underset{(2r)\times(2r-1)}{S}$$

“Direct” candidate ADP obfuscator for **NC^1**

Step 2: Recursively obfuscate sub-circuits

2b: $\mathbf{f(X) = g(X)} \vee \mathbf{h(X)}$:

$$\mathbf{A^f(X)} = \underset{(2r)\times(2r)}{\mathbf{R}} \begin{bmatrix} \mathbf{A^g(X)} & \mathbf{T(x)} \\ \mathbf{0} & \mathbf{A^h(X)} \end{bmatrix} \underset{(2r)\times(2r)}{\mathbf{S}}$$

Some special cases

Point functions

$$\underset{1 \times n}{\mathbf{R}} \begin{bmatrix} A^1(X_1) & 0 & 0 & 0 \\ 0 & A^2(X_2) & 0 & 0 \\ 0 & 0 & A^3(X_3) & 0 \\ 0 & 0 & 0 & A^4(X_4) \end{bmatrix} \underset{n \times 1}{\mathbf{S}}$$

Point functions

To obfuscate a point **y**:

$$\mathbf{A(X) = A_0 + A_1\, X_1 + \ldots + A_n\, X_n}$$

$\mathbf{A_1, \ldots, A_n}$ random scalars

$$\mathbf{A_0 = -\, A_1 y_1 - \ldots - A_n y_n}$$

Point functions

$\mathbf{A_1, \ldots, A_n}$ random scalars $\quad \mathbf{A_0 = -A_1 y_1 - \ldots - A_n y_n}$

Security: small field + entropic $\mathbf{y} \rightarrow \mathbf{A_0}$ is random
$\mathbf{y}$ actually information-theoretically hidden
finding (some) accepting $\mathbf{y}$ = 1D-SIS

Problem: small field means correctness errors
Problem: over even moduli, random parity of $\mathbf{y}$ revealed

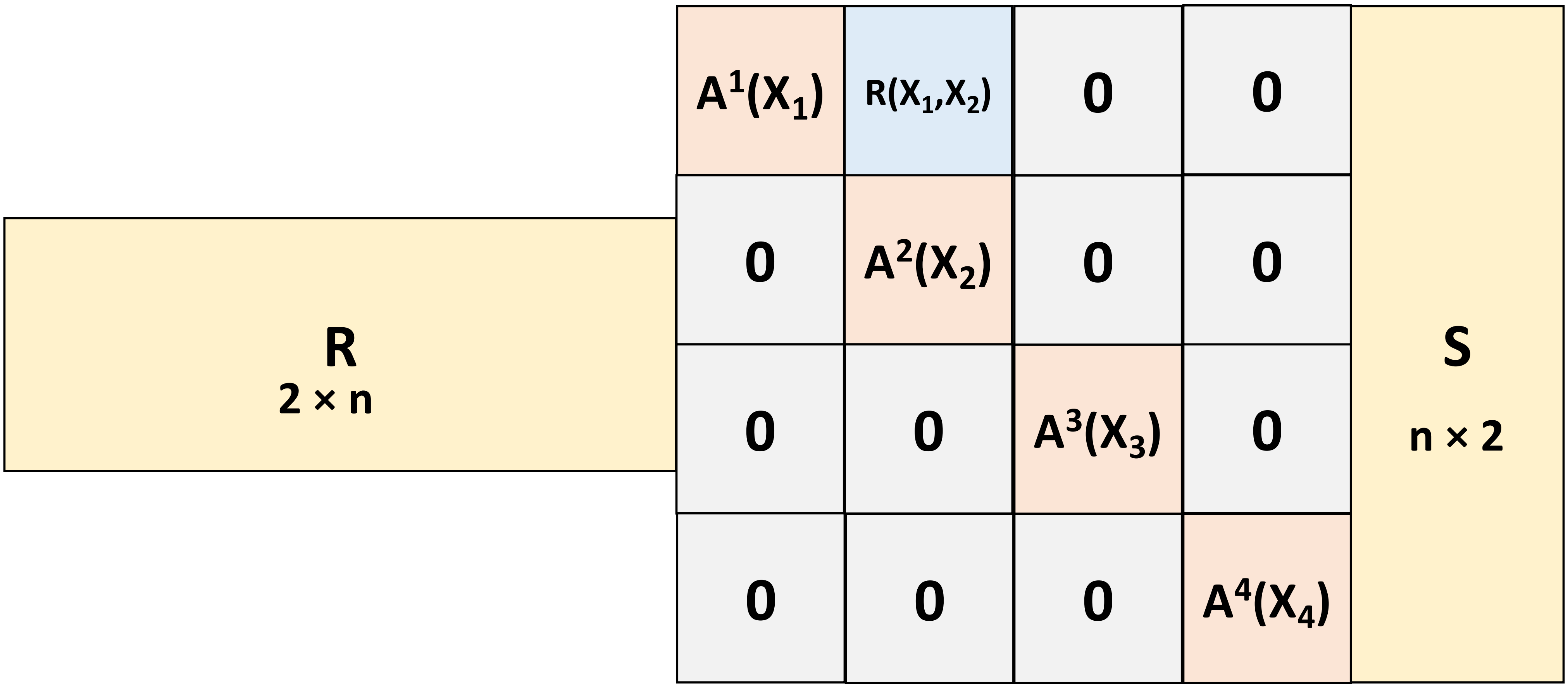
Point functions

$\mathbf{A_1, \ldots, A_n}$ random scalars $\quad \mathbf{A_0 = - A_1 y_1 - \ldots - A_n y_n}$

Security: large field insecure

$\mathbf{(1,y)}$ unique short vector orthogonal to $\mathbf{(A_0, A_1, \ldots)}$

Point functions, but with an OR gate



Point functions, but with an OR gate

Security: OR gate location trivially revealed

Coefficient of wires involved in OR gate
has rank **2**, all other matrices have rank **1**

Positive result

Thm [Bartusek-Lepoint-Ma-**Z**’19]:

“secure” ADPs for conjunctions under LPN

Witness Encryption

Witness Encryption

Enc(x, b) → c

Dec(x, w, c) → b

If **x** is false, **Enc(x,0)** $\approx_c$ **Enc(x,1)**

Candidate Witness Encryption

Assume **x** is (NP complete) *vector subset sum* instance

$$\mathbf{x} = \exists \mathbf{w} \in \{\mathbf{0,1}\}^n \text{ s.t. } \mathbf{M.w = t} \text{ (over } \mathbb{Z}\text{)}$$

Enc(x,0): random ADP. Det always non-zero

Enc(x,1): $\mathbf{A(X)} = \sum_i \mathbf{(M.X\text{-}t)_i\, R_i}$ $\mathbf{R_i}$ random matrices

So far, insecure:

Evaluation also works for **w** that are not **{0,1}**

Candidate Witness Encryption

$$\mathbf{Enc(x,1): A(X) = \textstyle\sum_i (M.X\text{-}t)_i\, R_i + B(X)}$$

$$\mathbf{Det(B(X))=0} \text{ iff } \mathbf{X} \text{ in } \mathbf{\{0,1\}^n}$$

Ex: Choose random **B** s.t.

$$\begin{aligned}
\mathbf{0} &\mathbf{= (1,X)\,.\,B(X)} \\
&\mathbf{= (1,X)\,.\,B_0 + (1,X)\,.\,B_1X_1 + \ldots} \\
&\mathbf{= (1,X_1,X_2,\ldots)\,.\,B_0 + (X_1,X_1^2,X_1X_2,\ldots)\,.\,B1 + \ldots} \\
&\mathbf{= (1,X_1,X_2,\ldots)\,.\,B_0 + (X_1,X_1\,,X_1X_2,\ldots)\,.\,B1 + \ldots}
\end{aligned}$$

Since **X** is binary