Dot-product: Vectors over $GF(2)$

Consider the dot-product of 11111 and 10101:

$$
\begin{array}{ccccccccccc}
 & 1 & & 1 & & 1 & & 1 & & 1 & & \\
\bullet & 1 & & 0 & & 1 & & 0 & & 1 & & \\
\hline
 & 1 & + & 0 & + & 1 & + & 0 & + & 1 & = & 1 \\
 & 1 & & 1 & & 1 & & 1 & & 1 & & \\
\bullet & 1 & & 0 & & 1 & & 0 & & 1 & & \\
\hline
 & 0 & + & 0 & + & 1 & + & 0 & + & 1 & = & 0
\end{array}
$$

Dot-product: Simple authentication scheme

- Usual way of logging into a computer with a password is subject to hacking by an eavesdropper.
- **Alternative:** Challenge-response system
  - Computer asks a question about the password.
  - Human sends the answer.
  - Repeat a few times before human is considered authenticated.

  Potentially safe against an eavesdropper since probably next time will involve different questions.
- Simple challenge-response scheme based on dot-product of vectors over $GF(2)$:
  - Password is an n-vector $\hat{\mathbf{x}}$.
  - Computer sends random n-vector $\mathbf{a}$
  - Human sends back $\mathbf{a} \cdot \hat{\mathbf{x}}$.

Dot-product: Simple authentication scheme

- **Example:** Password is $\hat{\mathbf{x}} = 10111$.
- Computer sends $\mathbf{a}_1 = 01011$ to Human.
- Human computes dot-product
  $\mathbf{a}_1 \cdot \hat{\mathbf{x}}$:

$$\begin{array}{cccccccccccc} & 0 & & 1 & & 0 & & 1 & & 1 & & \\ \bullet & 1 & & 0 & & 1 & & 1 & & 1 & & \\ \hline & 0 & + & 0 & + & 0 & + & 1 & + & 1 & = & 0 \end{array}$$

and sends $\beta_1 = 0$ to Computer.

Dot-product: Attacking simple authentication scheme

How can an eavesdropper Eve cheat?

- She observes a sequence of challenge vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m$ and the corresponding response bits $\beta_1, \beta_2, \ldots, \beta_m$.
- Can she find the password?

She knows the password must satisfy the linear equations

$$\begin{aligned} \mathbf{a}_1 \cdot \mathbf{x} &= \beta_1 \\ \mathbf{a}_2 \cdot \mathbf{x} &= \beta_2 \\ &\vdots \\ \mathbf{a}_m \cdot \mathbf{x} &= \beta_m \end{aligned}$$

Questions:

- How many solutions?
- How to compute them?

Answers will come later.

Dot-product: Attacking simple authentication scheme

Another way to cheat?

Can Eve derive a challenge for which she knows the response?

Algebraic properties of dot-product:

- **Commutativity:** $\mathbf{v} \cdot \mathbf{x} = \mathbf{x} \cdot \mathbf{v}$
- **Homogeneity:** $(\alpha\, \mathbf{u}) \cdot \mathbf{v} = \alpha\, (\mathbf{u} \cdot \mathbf{v})$
- **Distributive law:** $(\mathbf{v}_1 + \mathbf{v}_2) \cdot \mathbf{x} = \mathbf{v}_1 \cdot \mathbf{x} + \mathbf{v}_2 \cdot \mathbf{x}$

Example: Eve observes

- challenge 01011, response 0
- challenge 11110, response 1

$$
\begin{array}{rcccc}
(01011 + 11110) \cdot \mathbf{x} & = & 01011 \cdot \mathbf{x} & + & 11110 \cdot \mathbf{x} \\
 & = & 0 & + & 1 \\
 & = & & 1 &
\end{array}
$$

For challenge $01011 + 11110$, Eve can derive right response.

Dot-product: Attacking simple authentication scheme

More generally, if a vector satisfies equations

$$\begin{aligned} \mathbf{a}_1 \cdot \mathbf{x} &= \beta_1 \\ \mathbf{a}_2 \cdot \mathbf{x} &= \beta_2 \\ &\vdots \\ \mathbf{a}_m \cdot \mathbf{x} &= \beta_m \end{aligned}$$

then what other equations does the vector satisfy?
Answer will come later.