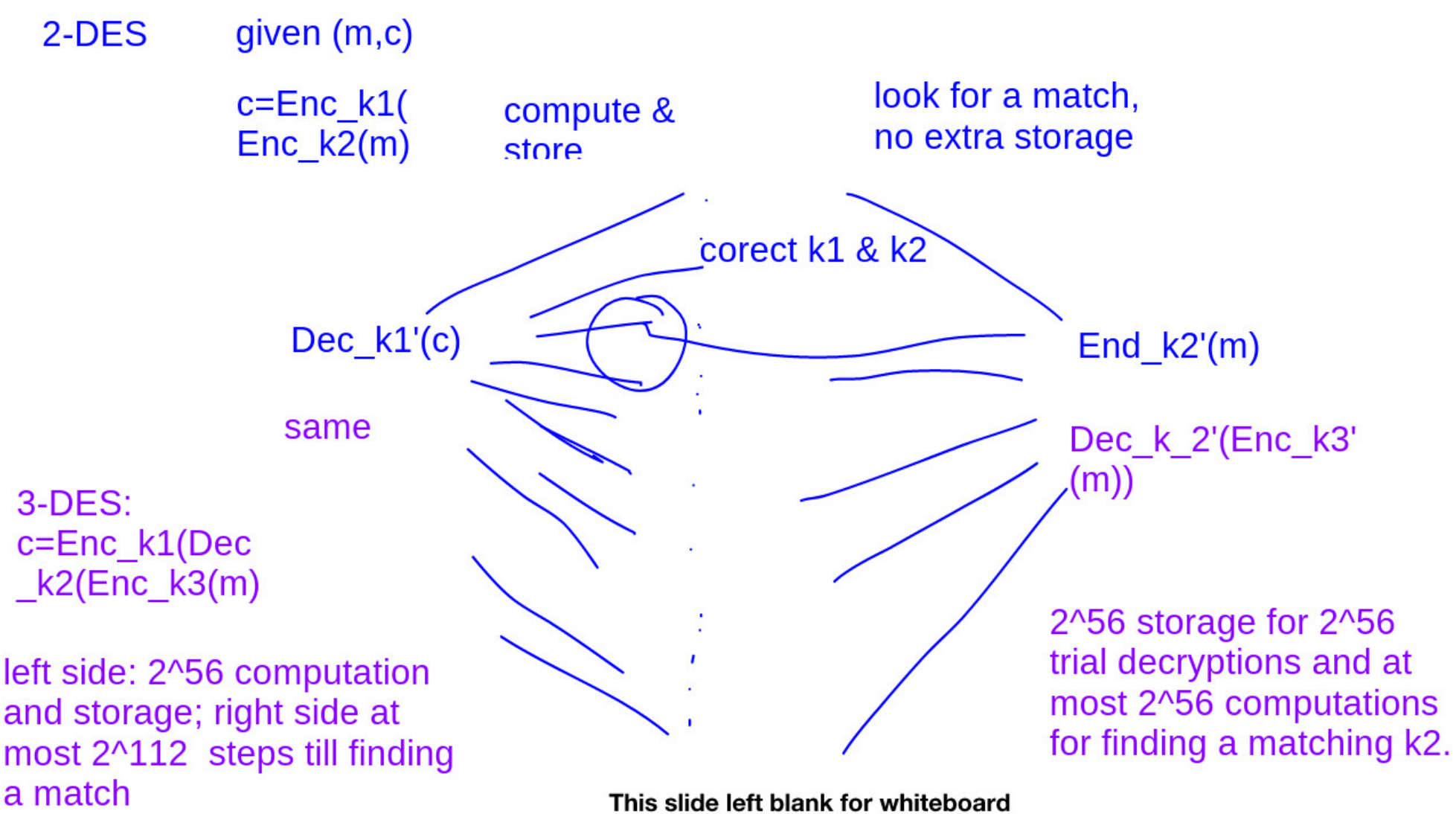
Live session 07 Dec 2020

Tanja Lange

Eindhoven University of Technology

2WF80: Introduction to Cryptology

Picture from Canvas conference session



This slide left blank for whiteboard

Picture from Canvas conference session

Generalize to 2n+1-DES. Then left hand side does 2^{56*n} operations and stores them. The RHS does 2^{{n+1)56} operations

Grover's seach algorithm take 2^{n/2} to compute a key of n bits.

However this needs a very big and stable quantum computer.

Padding is often dangerous; look up padding attacks.
Padding is encoding (not encryption or such), the operation is public and invertible. Communicating parties need to agree on how it is used.

Merkle-Damgård construction

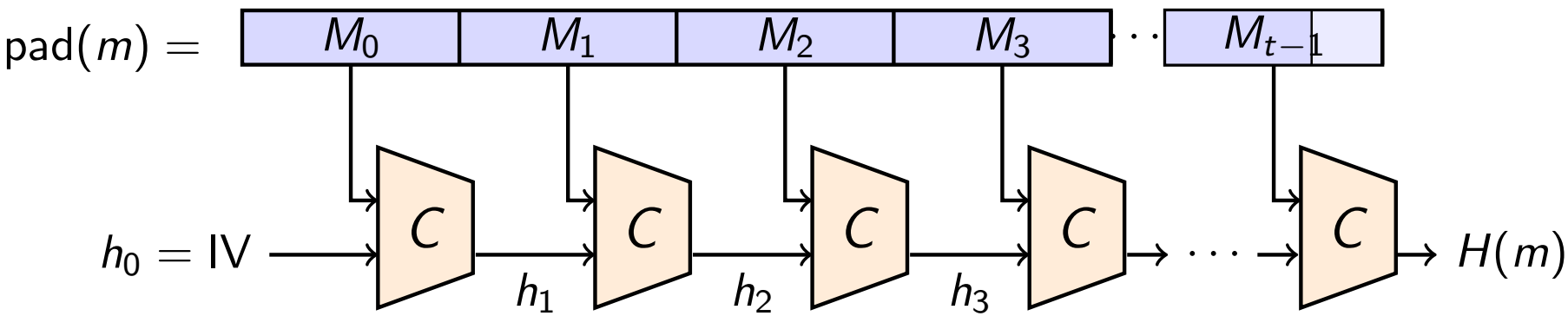
While the definition says $H : \{0,1\}^* \to \{0,1\}^n$
most constructions take data in blocks of a fixed number of bits.

Let $\text{pad}(m) = M_0\, M_1\, M_1\, \ldots\, M_{t-1}$ be the message padded up to a multiple of the block length n so that $m = m_0\, m_1\, m_2 \ldots m_{\ell-1}$ turns into
$M_0 = m_0\, m_1\, m_2 \ldots m_{n-1}, M_1 = m_n\, m_{n+1}\, m_{n+2} \ldots m_{2n-1}, \ldots$
$M_{t-1} = m_{(t-1)n}\, m_{(t-1)n+1}\, m_{(t-1)n+2} \ldots m_{\ell-1}\, p_0\, p_1\, \ldots\, p_{j-1}$, where
$t = \lceil \ell/n \rceil$, $p_0, p_1, \ldots, p_{j-1}$ are padding bits and $j = tn - \ell$

Merkle-Damgård construction

While the definition says $H : \{0,1\}^* \to \{0,1\}^n$
most constructions take data in blocks of a fixed number of bits.

Let $\text{pad}(m) = M_0\, M_1\, M_1\, \ldots\, M_{t-1}$ be the message padded up to a multiple of the block length n so that $m = m_0\, m_1\, m_2 \ldots m_{\ell-1}$ turns into
$M_0 = m_0\, m_1\, m_2 \ldots m_{n-1}, M_1 = m_n\, m_{n+1}\, m_{n+2} \ldots m_{2n-1}, \ldots$
$M_{t-1} = m_{(t-1)n}\, m_{(t-1)n+1}\, m_{(t-1)n+2} \ldots m_{\ell-1}\, p_0\, p_1\, \ldots\, p_{j-1}$, where
$t = \lceil \ell/n \rceil$, $p_0, p_1, \ldots, p_{j-1}$ are padding bits and $j = tn - \ell$



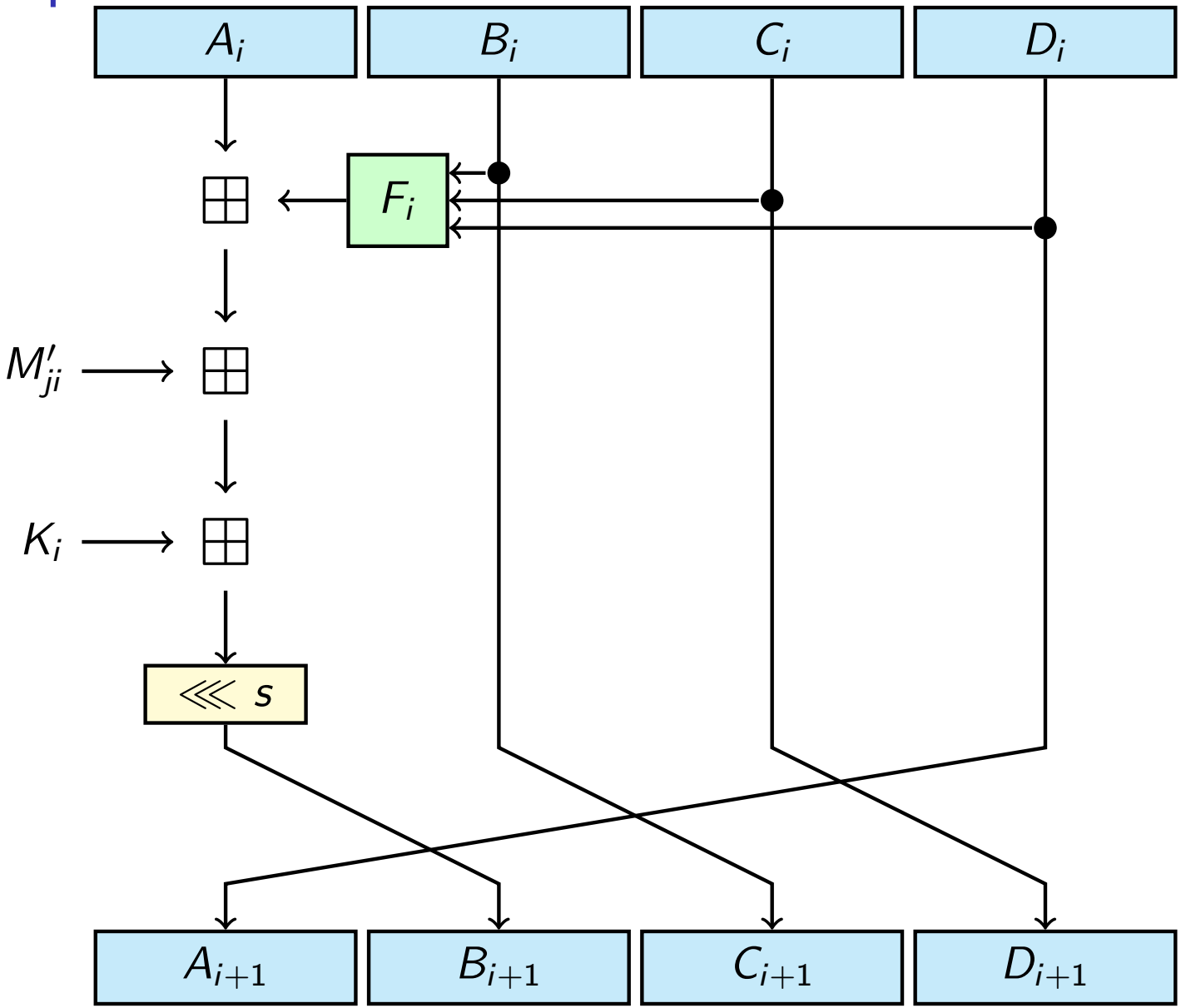
C in the Merkle-Damgård construction is a compression function

$$C : \{0,1\}^{2n} \to \{0,1\}^n.$$

Each step takes the n-bit h_{i-1} (previous output or $h_0 = \text{IV}$) and n message bits and compresses these to $h_i = C(M_{i-1}, h_{i-1})$ of n bits.

Image credit: adapted from Jérémy Jean.

Example: MD4



Each M_j turns into 48 M_{ij} of 32-bit. One call to C is 48 rounds.

Picture credit Jérémy Jean