Hash Extension Attack Supplement

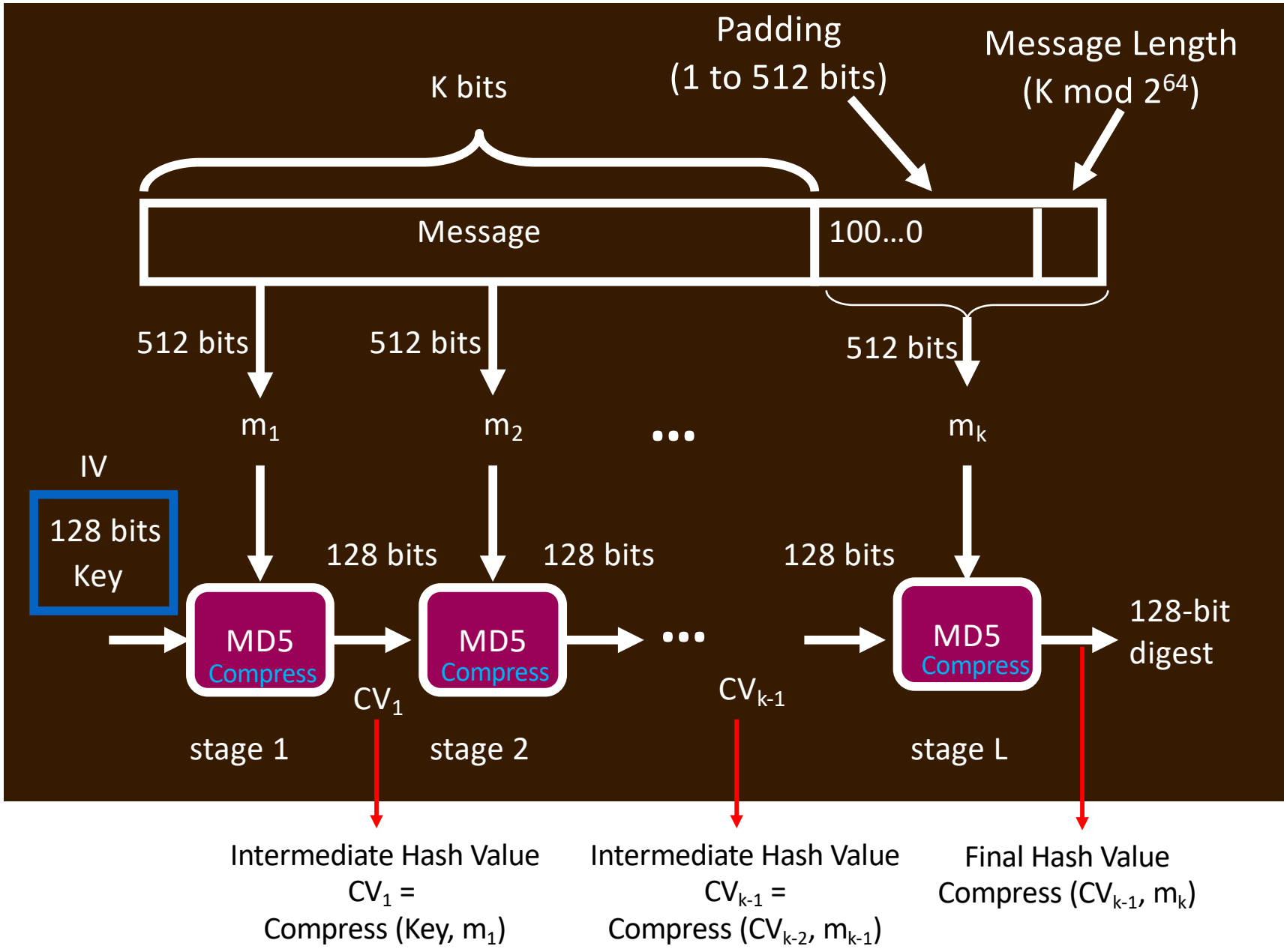
Kevin

Hash Length Extension

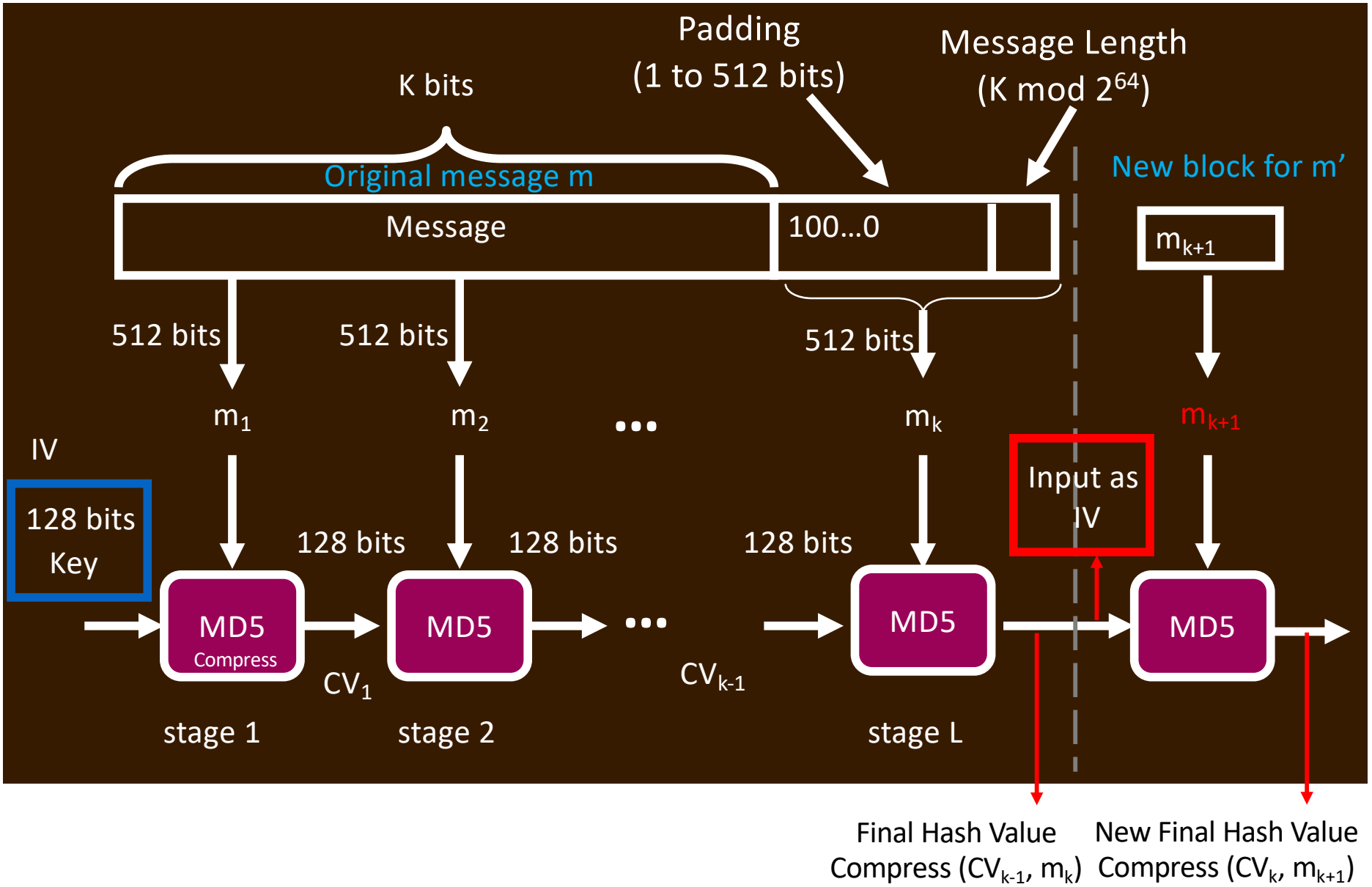
- Consider a message m is split into blocks m_1, m_2, …, m_k and hashed to a value h(m).
- Choose a new message m' that splits into the block m_1, m_2, …, m_k, m_{k+1} (the first k blocks are identical to m's).
- Therefore, h(m) is the intermediate hash value after k blocks in the computation of h(m').
- Thus, h(m') = `Compress`(h(m), m_{k+1}).
  - Construct a correctly padded m_{k+1}.

➔ Next slide for the detail

Basic Hash Operation for a Message m (split into blocks m_1, m_2, …, m_k)



Length Extension with a new block m_{k+1} after the original message m



Extension Attacks

- Given M1, and secret key K, can easily concatenate and compute the hash:
  H(K|M1|padding)
  - e.g., Alice sends an email M1 to Bob with a digest H(K|M1|padding)
  - Carol wants to add the message at the end of the email, saying M2, "P.S. Give Carol a promotion and triple her salary."
  - Given M1, H(K|M1|padding) from the email and M2, Carol should calculate H(K|M1|padding|M2|newpadding) but he doesn't know the key K.
  - Carol only has the email M1 and the digest H(K|M1|padding)

- However, given M1, M2, and H(K|M1|padding), it's easy to compute H(K|M1|padding|M2|newpadding) for some new message M2.

- Simply use H(K|M1|padding) as the IV for computing the hash of M2|newpadding
  - does not require knowing the value of the secret key K

➔ Next slide for the detail

Hash Extension Attack with a new message M2 after the original message M1