Formalizing Hash-then-Sign Signatures

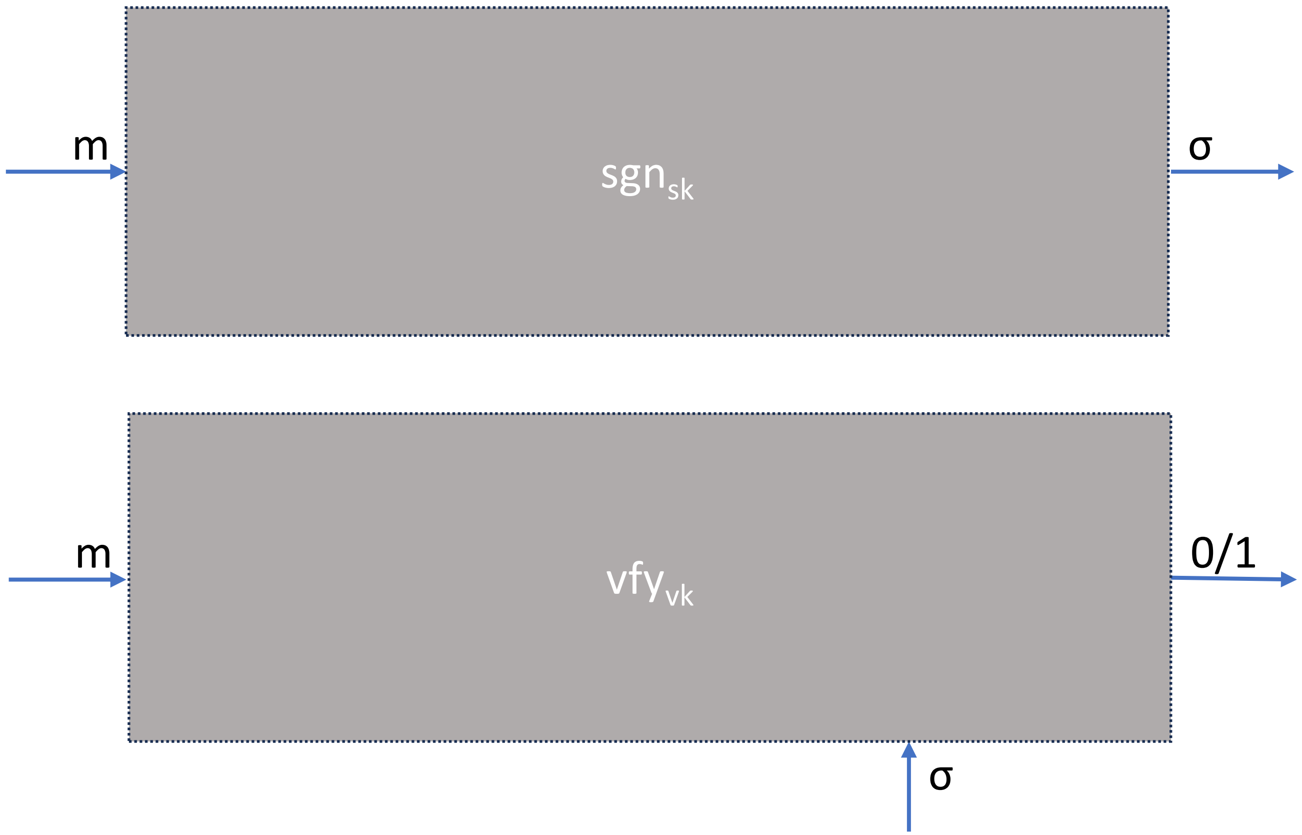
Bertram Poettering[1] and Simon Rastikian[1,2]

[1] IBM Research Europe, Zurich

[2] ETH Zurich

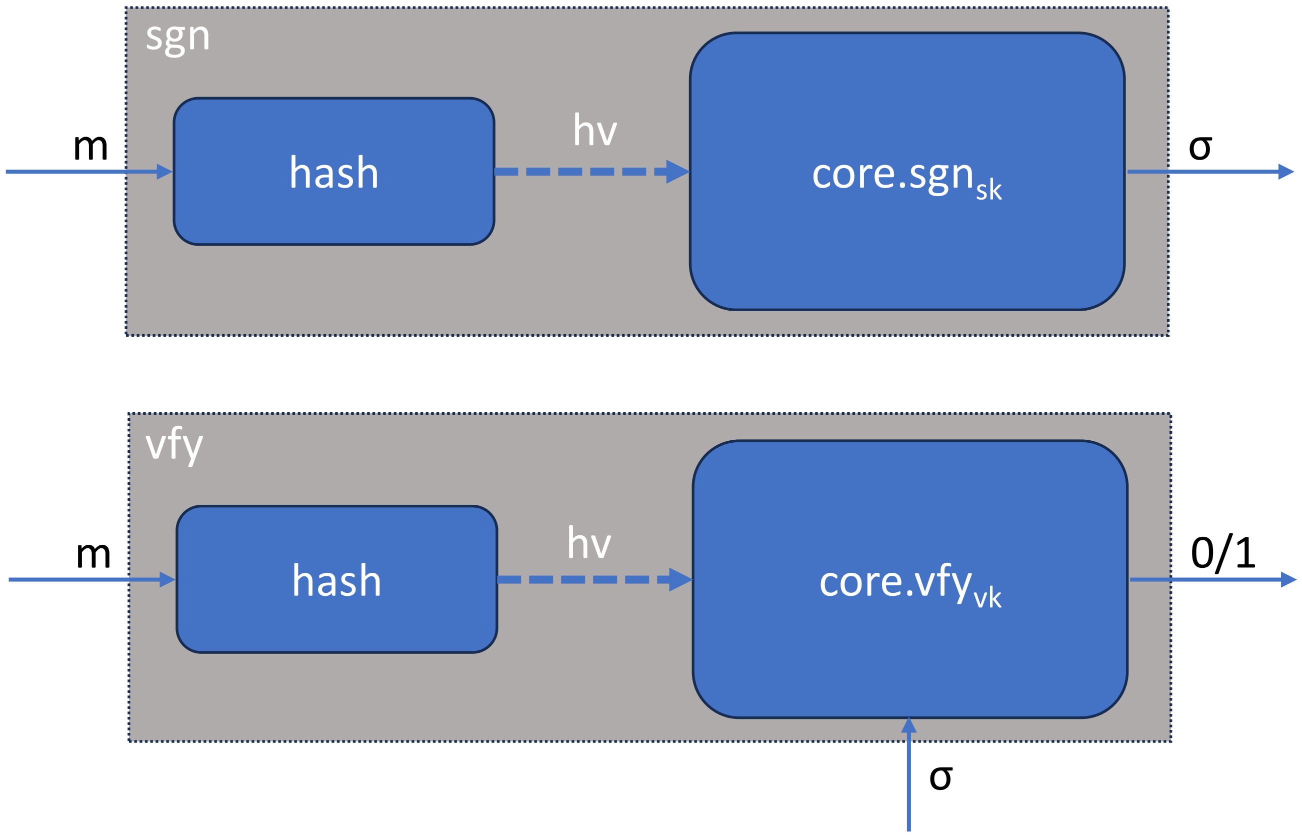
Digital Signature Schemes (DSS): Syntax

For simplicity, we omit the generation algorithm



Hash-then-Sign Structure

Engineering technique: signature and verification algorithms consist of two consecutive phases

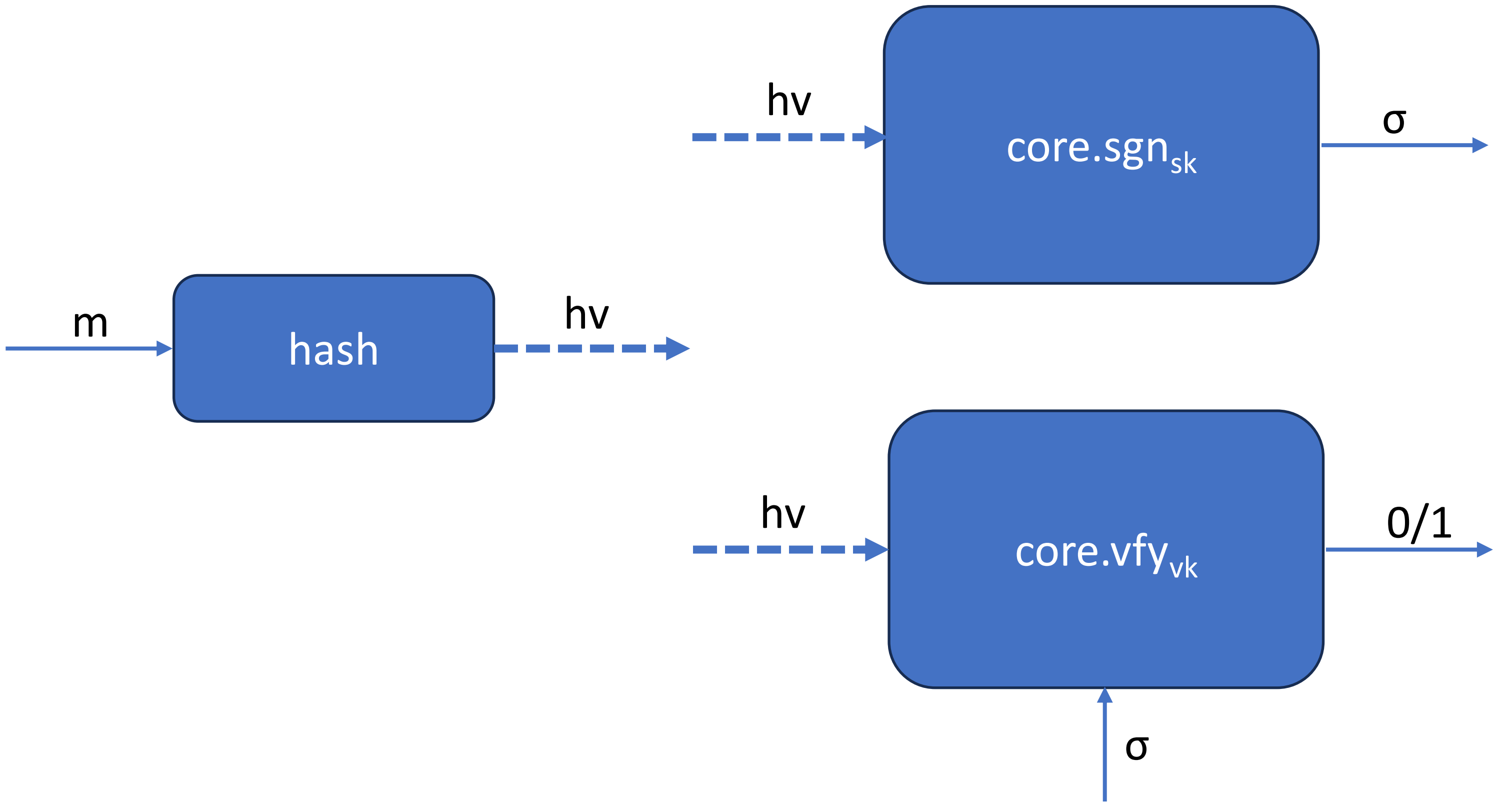


Common Digital Signature Schemes

Hash-then-Sign Signatures
PKCS#1v1.5
Full-Domain-Hash RSA
BLS signature scheme
ECDSA (American)
ECKCDSA (Korean)
GOST R 34.10-2012 (Russian)
SM2 (Chinese)

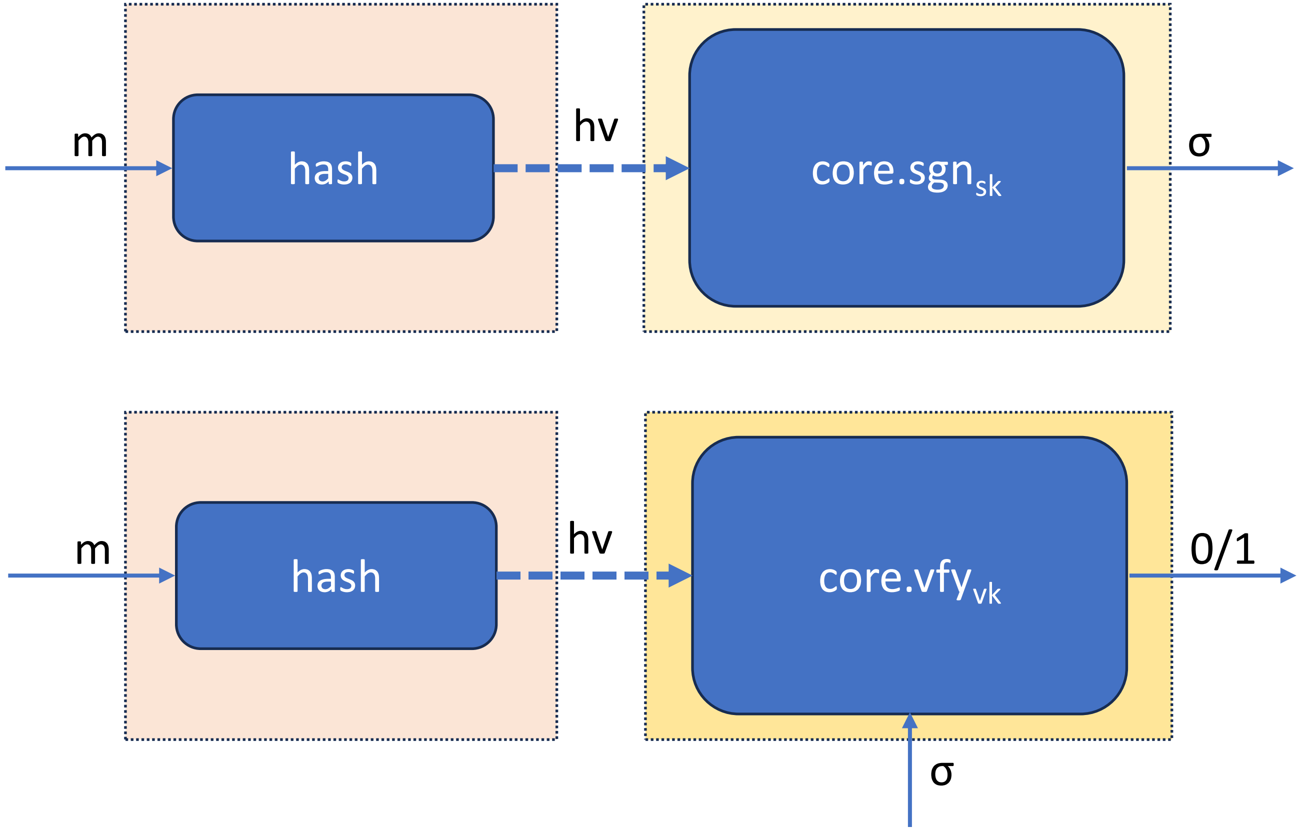
Non-Hash-then-Sign Signatures
SDSA
ECSDSA
BIP 340
ECFSDSA
EdDSA
RSA-PSS
SLH-DSA (SPHINCS+)

Hash-then-Sign Signatures: Terminology



Hash-then-Sign Signatures: Functionality

- Attractiveness of Hash-then-Sign Signatures: separating hash and core.sgn/vfy reflects different entities performing two tasks.



Separating hash and core.sgn: Relevance

1. Crypto libraries implementing dedicated API for separating the hashing and the core signing: Gcrypt, BoringSSL.

2. Standards organization support or are discussing the support of the separation of hashing and the core signing: PKCS#11, RFC8032, IETF/PQC forums

Hash-then-Sign: Application Examples

Core Routines	Message	Benefit of Separation
Provided by a Smartcard, HSM and TPM	Provided by a Host Computer	Optimizing Space and Speed

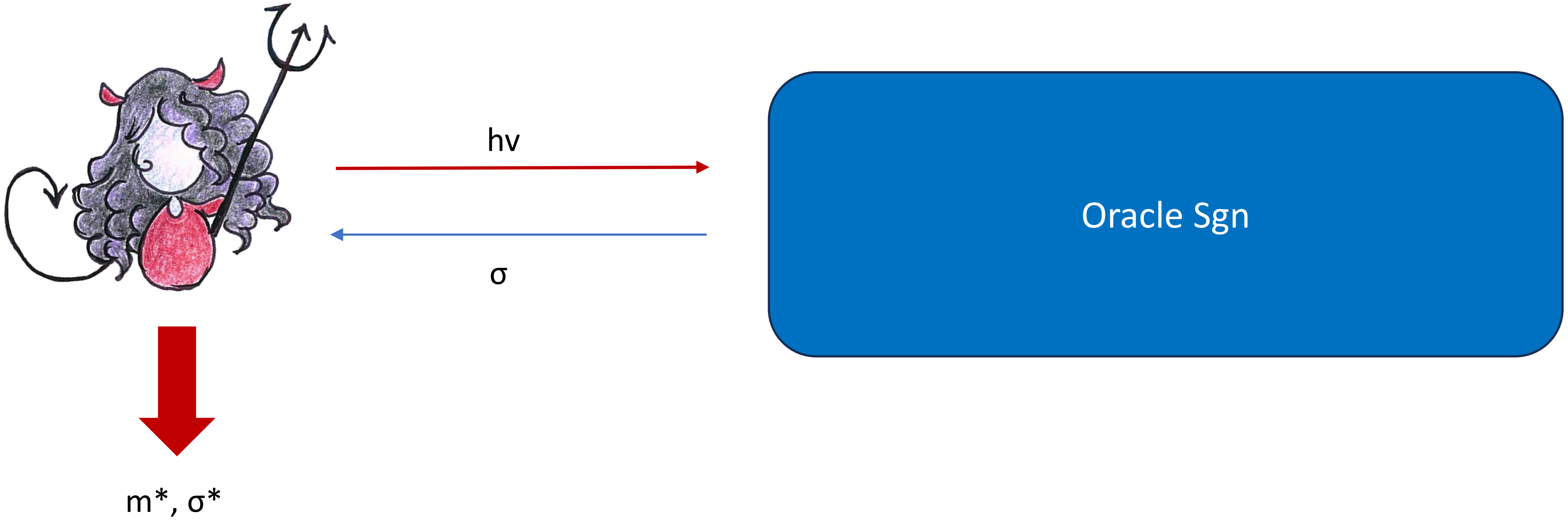
Core Routines	Message	Benefit of Separation
Complex implementation (big number arithmetic, optimized assembly instructions, side-channel attacks protection)	Provided by programs written in high level language where hash is optimized (SHA2 in Python)	No copying long messages from high-level applications to low-level core

Hash-then-Sign Signatures: Security

We look into the security of the Hash-then-Sign schemes when the hash function is separated from the core signature. More precisely when the hashing is malicious.

Hash-then-Sign Signatures	Security
PKCS#1v1.5	?
Full-Domain-Hash RSA	?
BLS signature scheme	?
ECDSA (American)	?
ECKCDSA (Korean)	?
GOST R 34.10-2012 (Russian)	?
SM2 (Chinese)	?

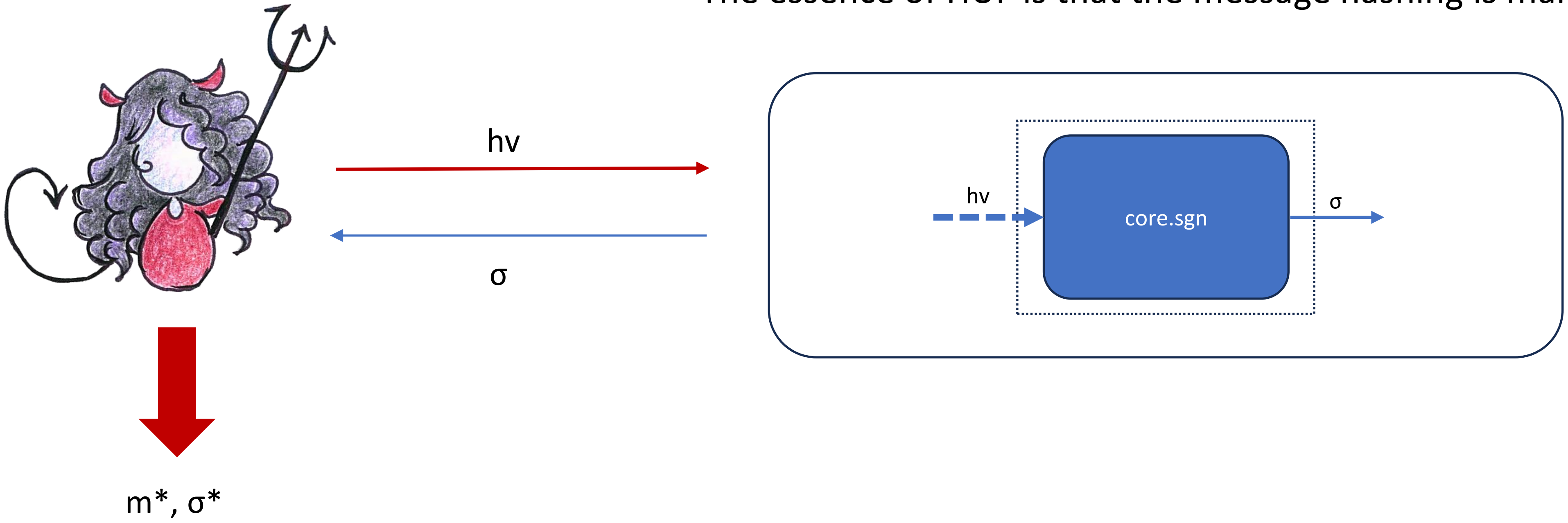
Hash-then-Sign Security Notion: HUF



Adversary wins if hv* ← hash(m*) is fresh and if the forgery is valid

Hash-then-Sign Security Notion: HUF

The essence of HUF is that the message hashing is malicious



Adversary wins if hv* ← hash(m*) is fresh and if the forgery is valid

HUF vs. UF

- Does the hashing make a difference?

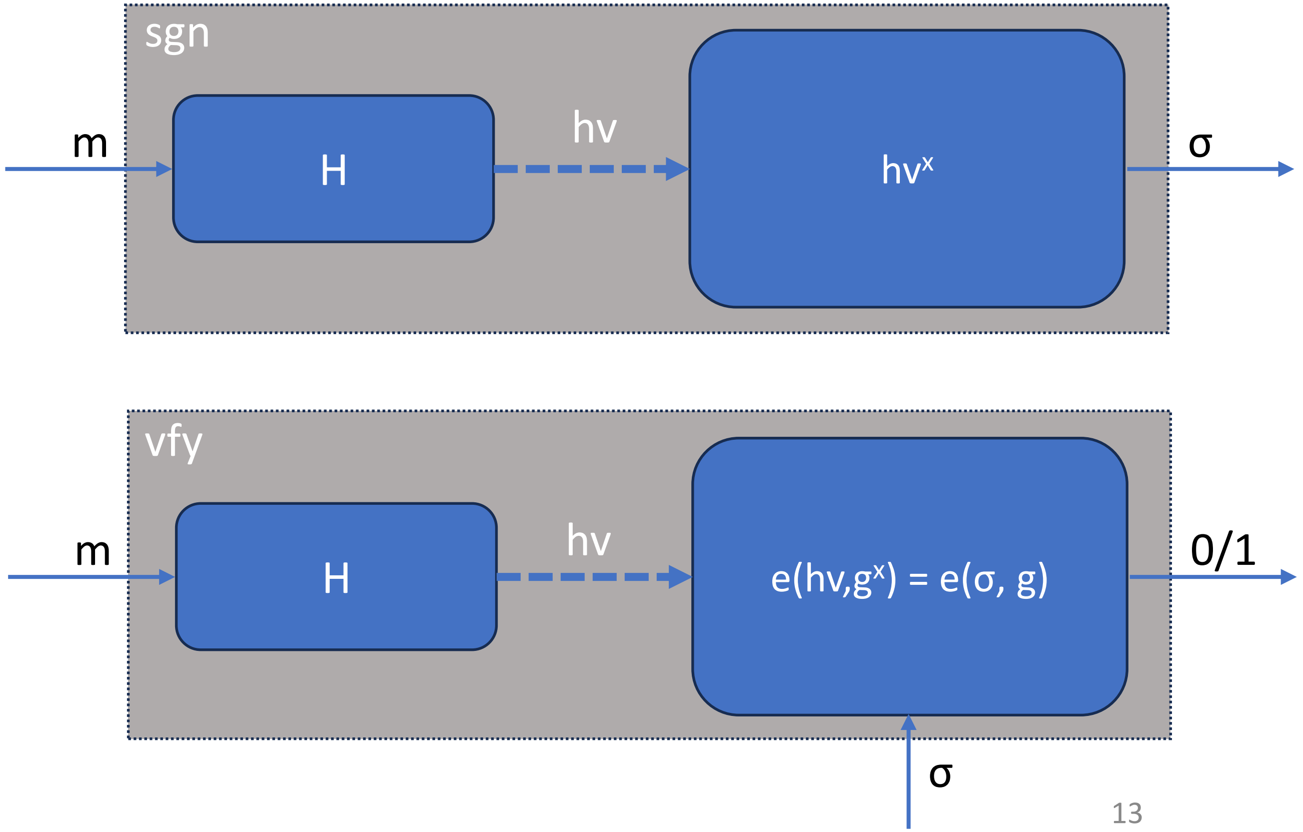
- We look into the relationship between HUF security and UF security:

HUF does not imply UF.

HUF vs. UF: Real-World Example

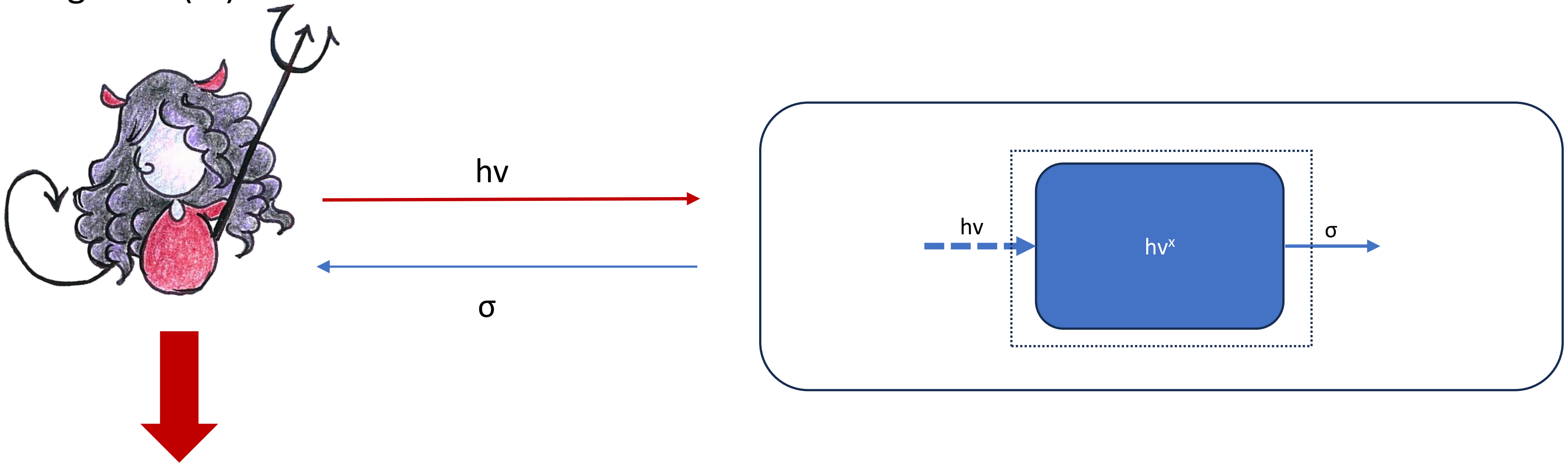
BLS Scheme: let G:= <g> be a cyclic group, H be a hash function from {0,1}* to G and e be a pairing.

BLS scheme is UF but not HUF



Insecurity of Hash-then-Sign BLS under HUF

Pick message m
Pick random exponent r
Blind hv:= g^r hash(m)



m, $\sigma/(g^x)^r$

Hash-then-Sign Signatures: Security

HtS-like Signatures	HUF security
PKCS#1v1.5	No
Full-Domain-Hash RSA	No
BLS signature scheme	No
ECDSA (American)	?
ECKCDSA (Korean)	?
GOST R 34.10-2012 (Russian)	?
SM2 (Chinese)	?

Investigating the Security of ECDSA

We reduce HUF to UF in the ECDSA case:

$$\mathrm{Adv}^{huf}(A) \leq \mathrm{Adv}^{uf}(B) + (6Q^2/|G|)$$

Many implementations separate the hash and core.sgn in ECDSA.

Good News: ECDSA is now proven to be HUF secure.

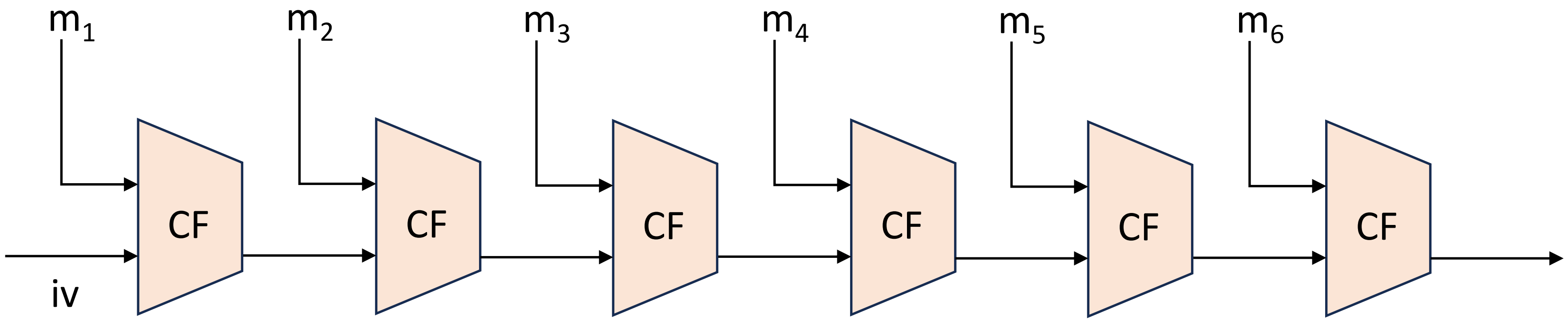
Hash-then-Sign Signatures: Security

HtS-like Signatures	HUF security
PKCS#1v1.5	No
Full-Domain-Hash RSA	No
BLS signature scheme	No
ECDSA (American)	Yes
ECKCDSA (Korean)	Yes
GOST R 34.10-2012 (Russian)	Yes
SM2 (Chinese)	Yes

A Generic Secure Method

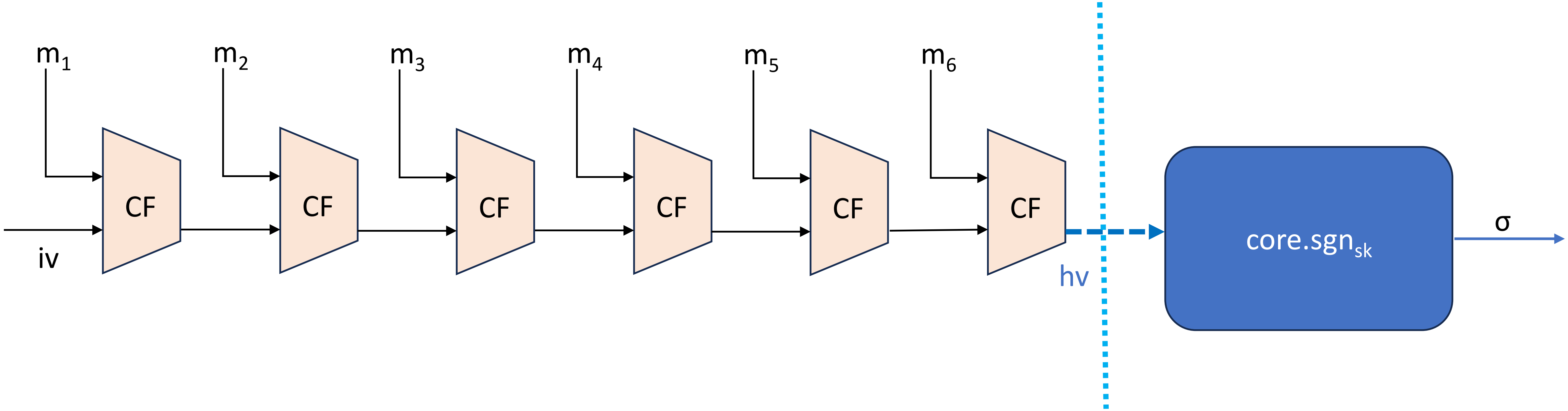
- We propose a generic method that allows a secure separation of the hashing and signing in Hash-then-Sign signature schemes.
- This method applies to all schemes of which the hash function is a Merkle-Damgård based construction.

Refresher: Merkle-Damgård Construction



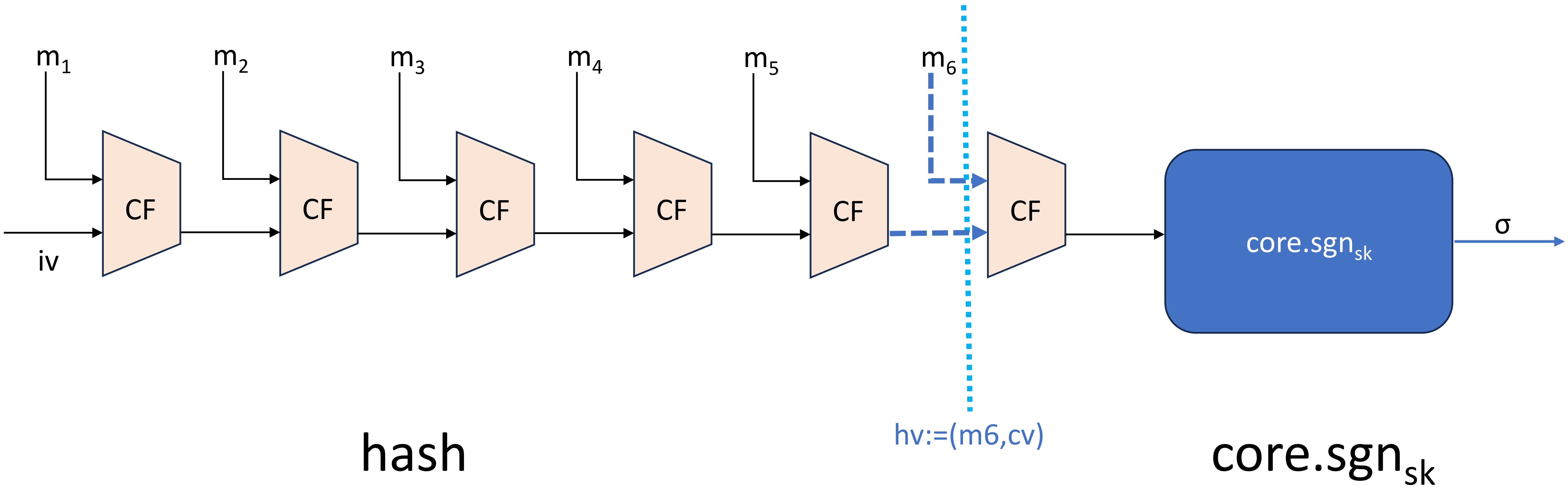
A Generic Secure Method

Usual approach: split completely the hashing phase from the core signing



A Generic Secure Method

Idea: Compute most of the hashing in hash except for the last CF. The core signing performs the last CF.



Conclusion

- We investigated the functionality of Hash-then-Sign signature.
- We introduced a new security model and studied real-world DSS in this model.

Future work: study the possibility of separation of the hashing and core signing for non-Hash-then-Sign signature schemes.

Thank you!