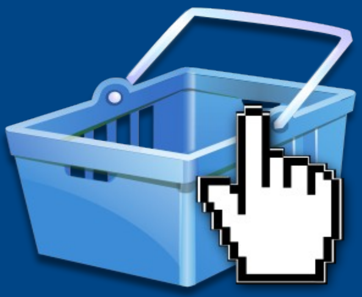
Tight Security

Dennis Hofheinz (KIT, Karlsruhe)

Motivation

- Real-world systems are multi-user, multi-use:


- Many instances of cryptographic building blocks used
  - For encryption schemes: many users (public keys), many ciphertexts
- Axiom: security means **all** instances secure

Motivation

- Example encryption schemes: we would want/expect

„**Only** way to break scheme in many-instance scenario is to factor 2000-bit number“

- What we currently have (for most existing systems):

„**Only** way break scheme in N-instance scenario is to factor (2000-f(N))-bit number“

(here, f(N) is somewhere in between O(log(N)) and O(N))

- ⇒ Need to know scenario size for bitlength recommendation!

Motivation

„**Only** way break scheme in n-instance scenario is to factor (2000-f(n))-bit number“

- Why is that so? (Why is it hard to get what we want?)
  - Maybe sometimes, attacks work better with many possible targets
  - Imagine a system in which each instance is just „bad“ with probability p (many instances ⇒ larger probability that one bad instance exists)

- „Social“ reason:
  - Easier to analyze 1-instance scenario
  - 1-instance security asymptotically implies n-instance scenario (loss n)

Game hops

- Closer look: technical difficulty in many-instance case
  - Typically, we proceed in game hops

(vector of) challenges

S Enc(se … c(random) A

<u>For this talk:</u>

cryptographic scheme tightly secure (under some assumption X)
$\Leftrightarrow$
security guarantees (given X) do not vanish in #instances
$\Leftrightarrow$
loss of security reduction to X does not depend on #instances

  - Star… attack
  - Slig… reduction)
  - Con… erall scheme
  - Typically, each step „costs" a reduction (and causes security loss)

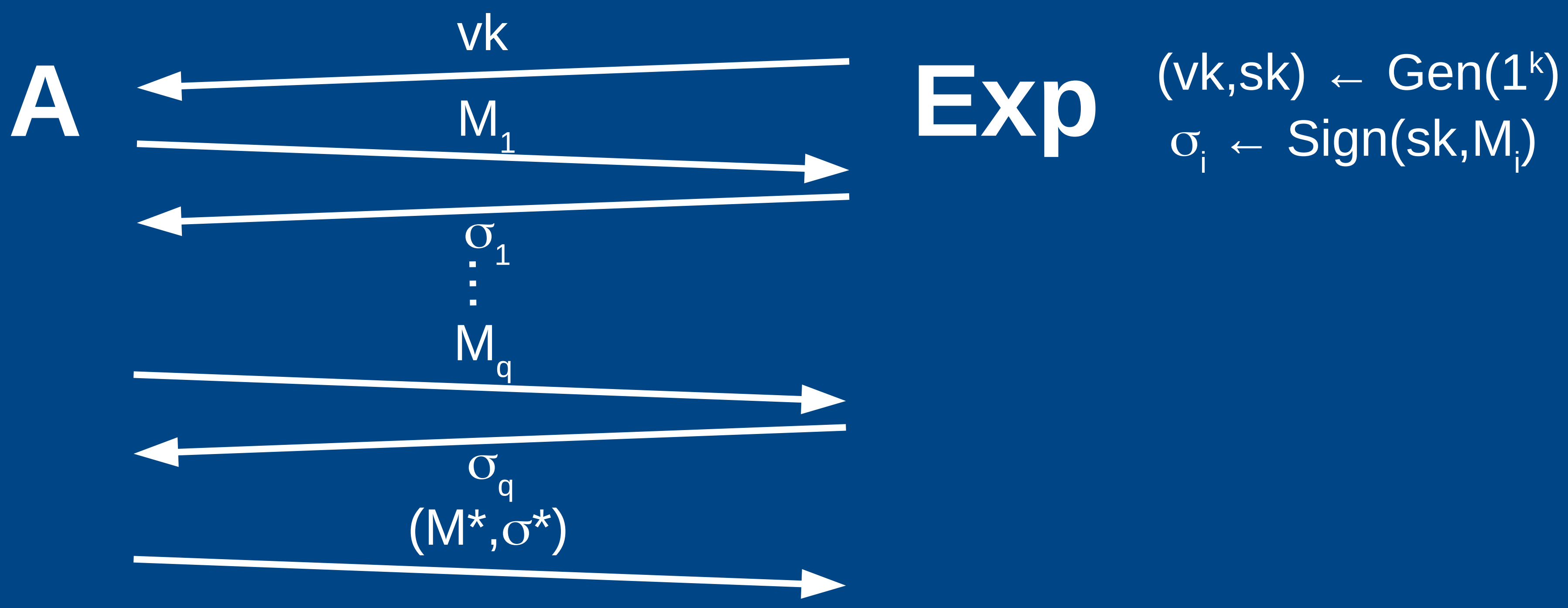
- **Difficulty:** keep number of steps/reductions low (constant)
  - **Substitute many challenges given to attacker in few steps**

Signature schemes

- Signature scheme (Gen,Sign,Ver) consists of 3 algorithms:
  - $Gen(1^k)$ generates a keypair (vk,sk)
  - Sign(sk,M) computes a signature σ for a message M
  - Ver(vk,σ,M) verifies whether a signature σ is valid for M
- Correctness: Ver(vk, Sig(sk, M), M) = 1 always

Technical goal: EUF-CMA

- Existential unforgeability under chosen-message attacks:



- Security ⇔ Pr[Ver(vk,M*,σ*) = 1 and M* fresh] negl. ∀ PPT A
- Scheme GS-compatible → many-instance encryption/signatures

Chen-Wee (Crypto 13)

- [CW13]-signatures for $M = (m_1, \ldots, m_n)$ are of the form

  $$(h_0, X \cdot \prod h_{i,mi})$$

  where X is part of the secret key, and $(h_0, h_{1,0}, h_{1,1}, \ldots, h_{n,0}, h_{n,1})$ chosen freshly from joint public distribution
- Strategy/goal of security proof:
  - Modify signatures given to **A** and accepted from **A** as valid:

    $$(h_0, RF(M) \cdot \prod h_{i,mi})$$
  - RF random function → **A**'s chance to find valid forgery negl.

Chen-Wee security proof

- [CW13]-proof gradually modifies definition of valid signatures:

$$(h_0, X\cdot\prod h_{i,mi}) \xrightarrow{\text{same as}} (h_0, RF(\varepsilon)\cdot\prod h_{i,mi})$$

n steps

$$\xrightarrow{\text{magic}} (h_0, RF(m_1)\cdot\prod h_{i,mi})$$

$$\xrightarrow{\text{magic}} (h_0, RF(m_1 m_2)\cdot\prod h_{i,mi})$$

$$\vdots$$

$$\xrightarrow{\text{magic}} (h_0, RF(m_1\cdots m_n)\cdot\prod h_{i,mi})$$

$$\xrightarrow{\text{same as}} (h_0, RF(M)\cdot\prod h_{i,mi})$$

- … ok, but what is that „magic“ step there?

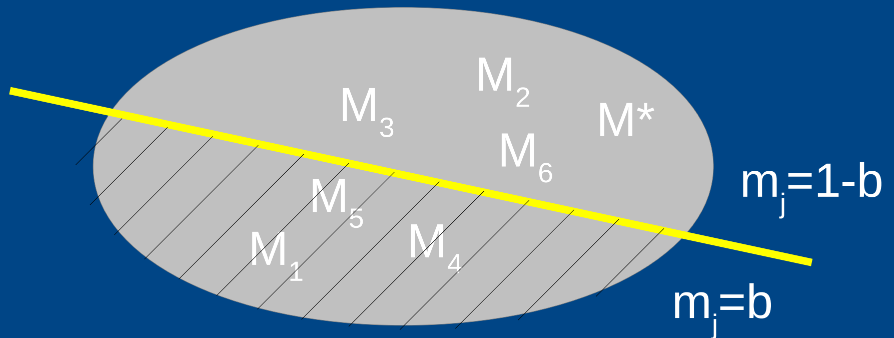
Chen-Wee high-level strategy

- Single hybrid step in [CW13]-proof:

$$(h_0, RF(m_1 \cdots m_{j-1}) \cdot \prod h_{i,mi}) \longrightarrow (h_0, RF(m_1 \cdots m_j) \cdot \prod h_{i,mi})$$

- How to get there in four easy steps: $(h_0, RF(m_1 \cdots m_{j-1}) \cdot \prod h_{i,mi})$

  1) Partition message space according to m_j:



Chen-Wee high-level strategy

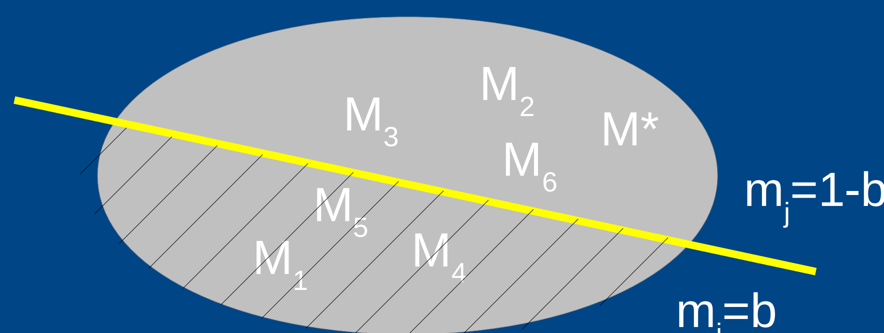
- Single hybrid step in [CW13]-proof:

$$(h_0, \mathrm{RF}(m_1 \cdots m_{j-1}) \cdot \prod h_{i,mi}) \longrightarrow (h_0, \mathrm{RF}(m_1 \cdots m_j) \cdot \prod h_{i,mi})$$

- How to get there in four easy steps: $(h_0, \mathrm{RF}(m_1 \cdots m_{j-1}) \cdot Z \cdot \prod h_{i,mi})$

freshly random, but only for m_j=b

1) Partition message space according to m_j:



2) Embed comp. challenge (only) into all $h_{i,b}$ for same random b

Chen-Wee high-level strategy

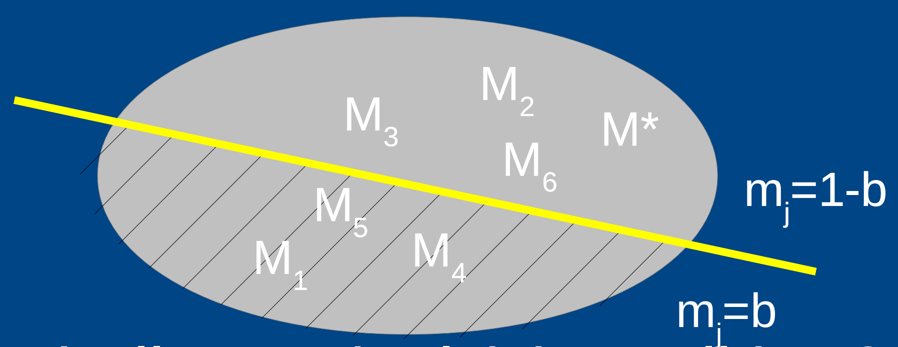
- Single hybrid step in [CW13]-proof:

  $(h_0, \text{RF}(m_1 \cdots m_{j-1}) \cdot \prod h_{i,mi}) \longrightarrow (h_0, \text{RF}(m_1 \cdots m_j) \cdot \prod h_{i,mi})$

- How to get there in four easy steps: $(h_0, \text{RF}(m_1 \cdots m_{j-1}) \cdot Z \cdot \prod h_{i,mi})$

  freshly random, but only for m_j=b

  1) Partition message space according to m_j:



  2) Embed comp. challenge (only) into all $h_{i,b}$ for same random b

  3) Hope that forgery M* has m_j=1-b (needed for verification)

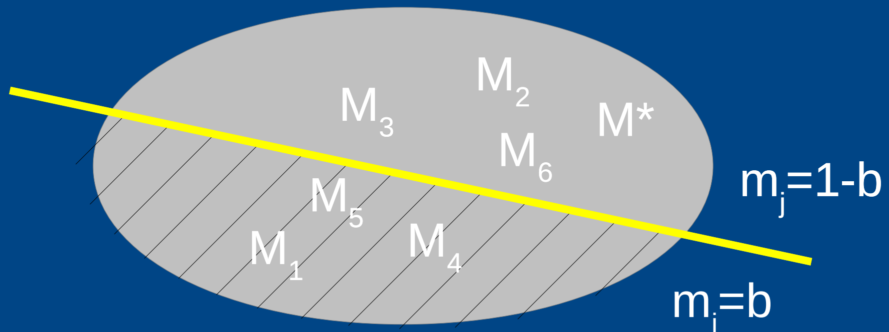
Chen-Wee high-level strategy

- Single hybrid step in [CW13]-proof:

  $(h_0, RF(m_1 \cdots m_{j-1}) \cdot \prod h_{i,mi}) \longrightarrow (h_0, RF(m_1 \cdots m_j) \cdot \prod h_{i,mi})$

- How to get there in four easy steps: $(h_0, RF(m_1 \cdots m_j) \cdot \prod h_{i,mi})$

  1) Partition message space according to m_j:



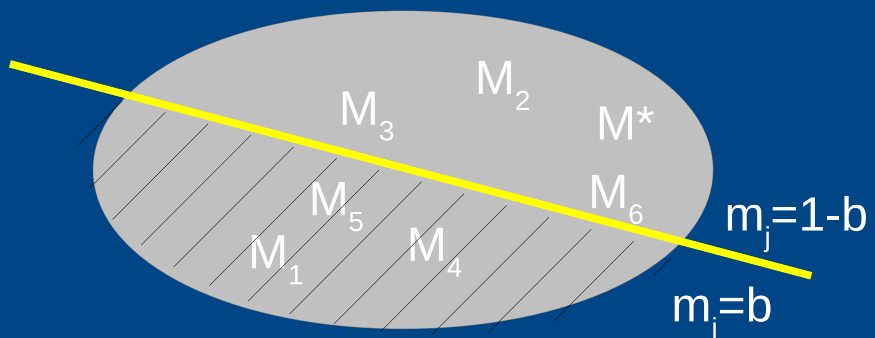
  2) Embed comp. challenge (only) into all $h_{i,b}$ for same random b

  3) Hope that forgery M* has m_j=1-b (needed for verification)

  4) Effect: added dependency on m_j in RF

Chen-Wee high-level strategy

- [CW13] require n = secpar hybrid steps → reduction loss is O(n)
- In each step, the message space is partitioned:



- Each of these partitions is prepared for in signature already:

$$(h_0, X\cdot\prod h_{i,mi})$$

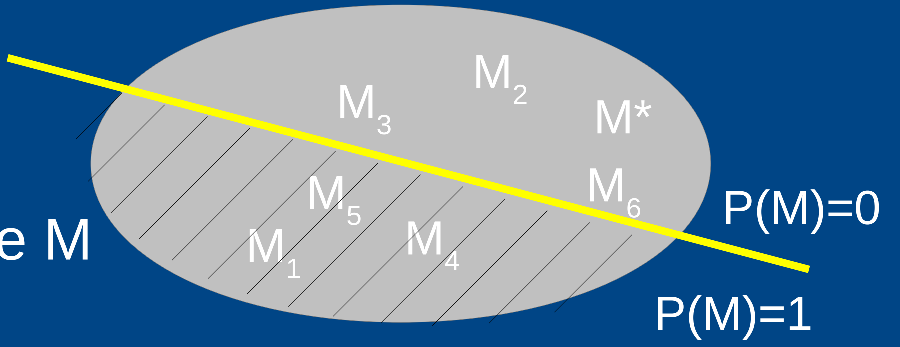
- **Consequence: O(n)-sized public pars** (that define $h_{i,b}$-dist.)
- **Note:** similar techniques exist in PRF context [NR97]

Variant: algebraic partitioning

- [H16]: implement [CW13] strategy with different partitioning

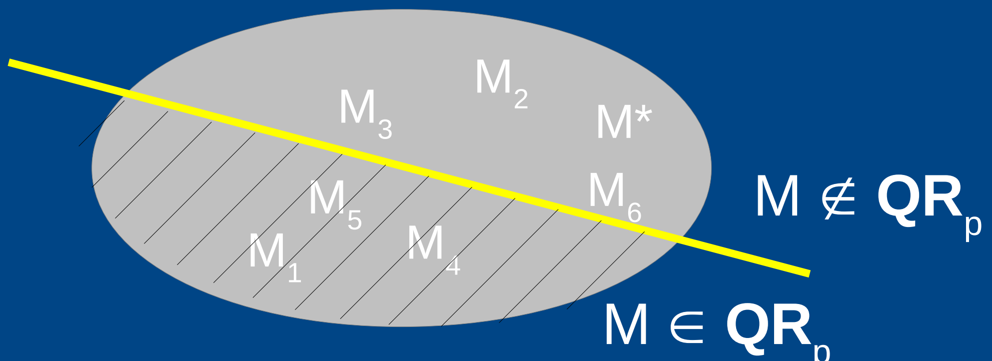
- **Specifically:** think of „more algebraic“ partitioning of messages
  - **Intuition:** more algebraic partitioning → partitioning can be „hidden“
  - **Hope:** not all partitionings used in proof have to be present in scheme

- So we're looking for a predicate P on messages such that…
  - P(M) = 1 for about half of all M
  - P itself is Groth-Sahai-compatible
  - Actually, we need many P that, taken together, uniquely identify a message M



Variant: algebraic partitioning

- **Predicate P:** Quadratic residuosity (modulo group order!)
  - Work in DDH group **G** of prime order p
  - Messages are $\mathbf{Z}_p$-elements (i.e., exponents)
  - Define P as $P(M) = 1 \Leftrightarrow M \in \mathbf{QR}_p \Leftrightarrow \exists\, r \neq 0$ with $r^2 = M \bmod p$



- **Problem:** provides only one partitioning of message space
- **Solution:** randomize P: set $P(M) = 1 \Leftrightarrow f(M) \in \mathbf{QR}_p$ for affine f

Corresponding signature scheme

- The verification key is $vk = (\text{ CRS}, pk, \text{Com}(f), \text{Com}(X) \text{ })$
- Signatures in „algebraic partitioning“ scheme are of the form:

$$(C{:=}\text{Enc}_{pk}(X), \pi_1, \pi_2)$$

- π_1 GS-NIZK for „know plaintext of C **or** $f(M) \in \mathbf{QR}_p$“
- π_2 GS-NIZK for „C encrypts X“ (simulated in proof, but **not sim.-snd.**)
- Proof gradually transforms signatures into:

$$(C{:=}\text{Enc}_{pk}(RF(M)), \pi_1, \pi_2)$$

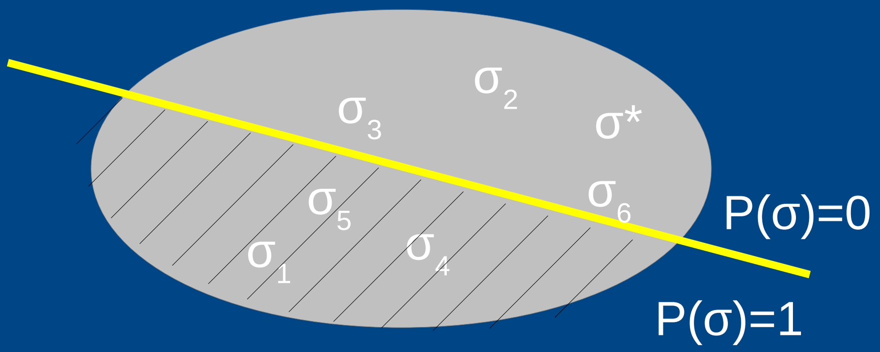
Variant: adaptive partitioning

- [H17,AHNOP17]: partition adaptively (i.e., predicate fixed in sig)
  - Essentially, P(σ) bit encrypted in σ
  - **Advantage:** compact signatures/keys, no quadratic $\mathbf{Z}_p$-equations
  - **Disadvantage:** switching the partitioning more complicated
    - During most of proof, necessary to decrypt P(σ*) to judge σ*
    - But: when switching partitioning, should not be able to decrypt P(σ)
    - Solution:

      (1) gradually randomize X→ F(M) in issued signatures, but

      (2) accept any X* that is a <u>reused</u> X=F(M) for an old M

      (3) use one-time sig. with key X



Encryption?

- These techniques also yield encryption schemes:
  - [CW13] actually IBE (variants lead to tightly IND-CCA secure PKE)
  - [H16,H17,GHK17] contain tightly IND-CCA secure PKE schemes

- Similar dilemma:
  - Security reduction needs to decrypt queries from adversary…
  - … but should be able to randomize many challenge ciphertexts

    → partition set of ciphertexts (not set of messages)

- Added difficulty: many decryptable, many non-decryptable C!
  - Solution: signatures/MACs → (DV-)NIZKs → Naor-Yung PKE

Related work

- So far, focus on Chen-Wee and own works, many others exist
  - PKE: [BBM00,HJ12,AKDNO13,LJYP14,GHKW16]
  - Sigs: [CMJ07,BKP14,LJYP14,S15,BKKP15,AHNOP17,GHKP18]
  - … and many more: (H)IBE, NIKE, AKE, NIZK, PRFs, …

- Surprisingly, similar technical problems/gadgets
  - Central: re-randomizability of DH-like assumptions

- (Largely) open: What about lattices?
  - … or adaptive corruptions?
  - … or other notion of scalability/tightness (e.g., memory)?

Last slide

Thanks for your attention!