Security Analysis of a 2/3-rate Double Length Compression Function in Black-Box Model

Wonil Lee (Speaker) – Kyushu University, Japan
Mridul Nandi – Indian Statistical Institute, India
Kouichi Sakurai – Kyushu University, Japan
Sangjin Lee – CIST, Korea University, Korea

Hash Function

- A hash function is a function from an arbitrary domain to a fixed domain.
- The hash function has been popularly used in digital signatures schemes, public key encryption, MAC etc.
- To have a good digital signature schemes or public key encryption, it is required that hash function should be **collision resistant** or **preimage resistant**.

Compression function

- Usually, one first design a fixed domain hash function (compression function) $f:\{0,1\}^{n+m} \rightarrow \{0,1\}^{n}$.
- And extend the domain to an arbitrary domain by iterating the compression function several times.
- The most popular method is known as MD-method.

To make the birthday attack infeasible

- Nowadays, people are interested in designing a bigger size hash function to make the birthday attack infeasible.

- One can do it by just constructing a compression function like SHA-512.

Our interest

- The other way is to construct it from a smaller size compression function.

  - In this case, one can study the security level of the bigger size hash function assuming some security level of underlying compression functions.

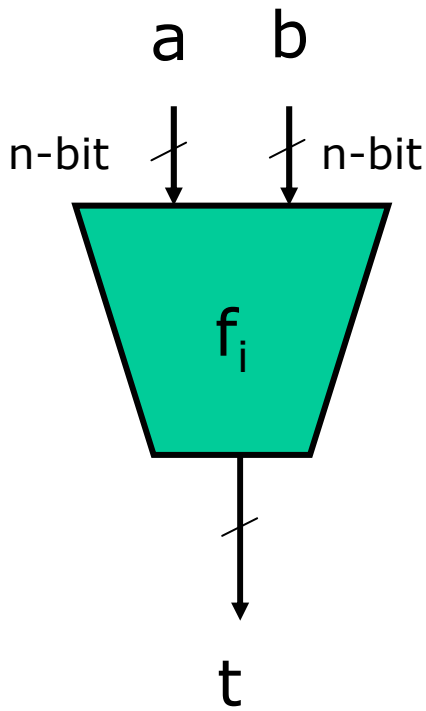
In this work

- If a single length compression function has output size n, then that of double length compression function is 2n.

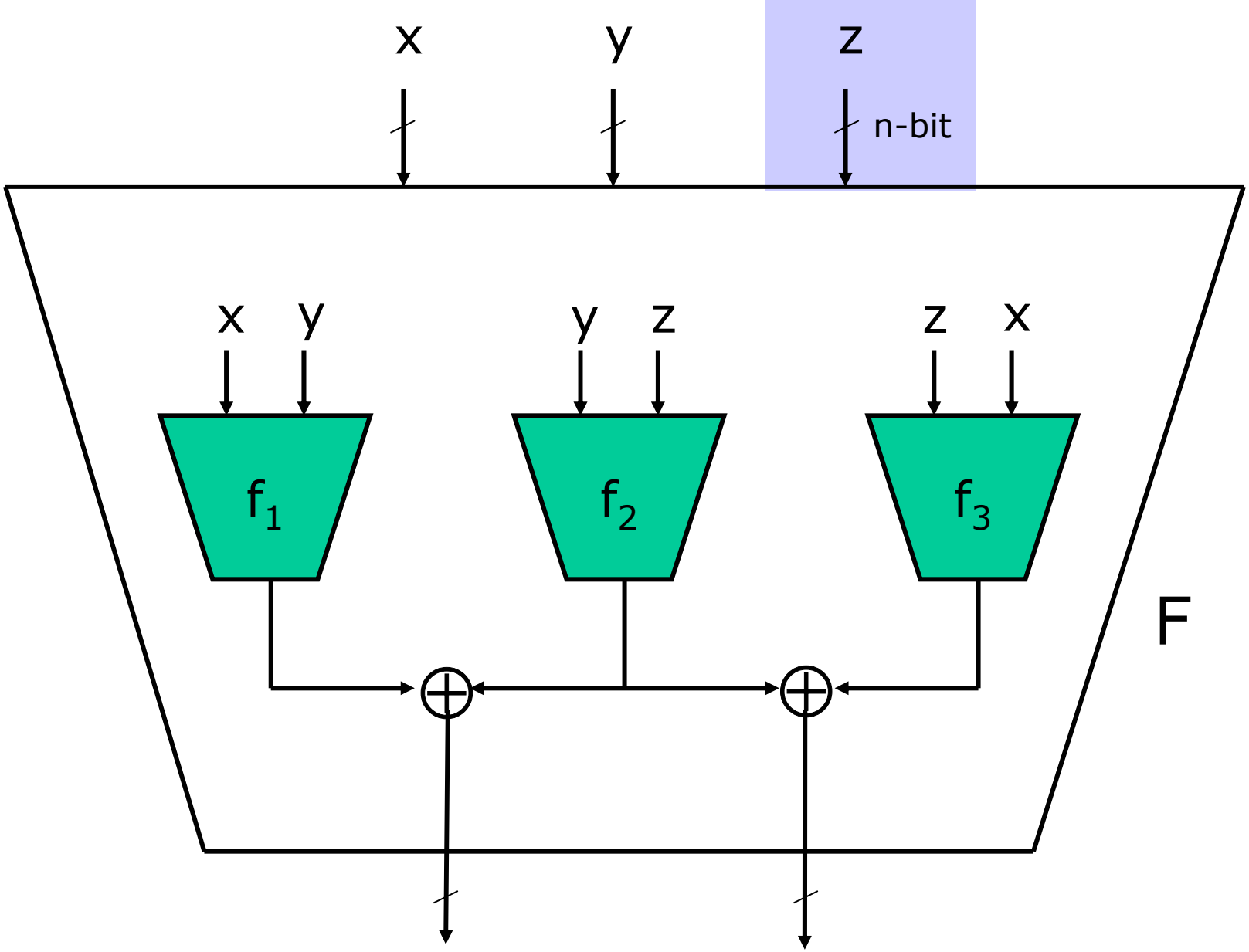
- In this work, in order to construct a double length compression function, we use **three invocations of independent single length compression functions or block ciphers** to hash **two message blocks**. Thus, the rate of the compression function is 2/3.

Construction

- If we have



i=1,2,3



A double length compression function [rate: 1/3]

Adversary - random oracle model

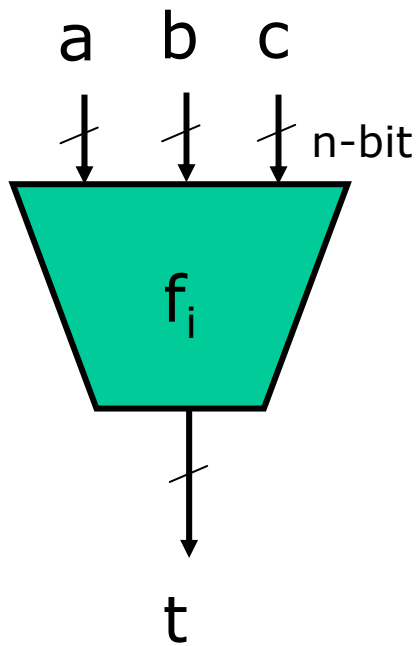
- Adversary can ask the oracles f_1, f_2, f_3.
- He can ask **(a, b)** to any one of the oracles f_1, f_2, f_3, and get a response **t** such that $f_i(a,b)=\mathbf{t}$.

Security

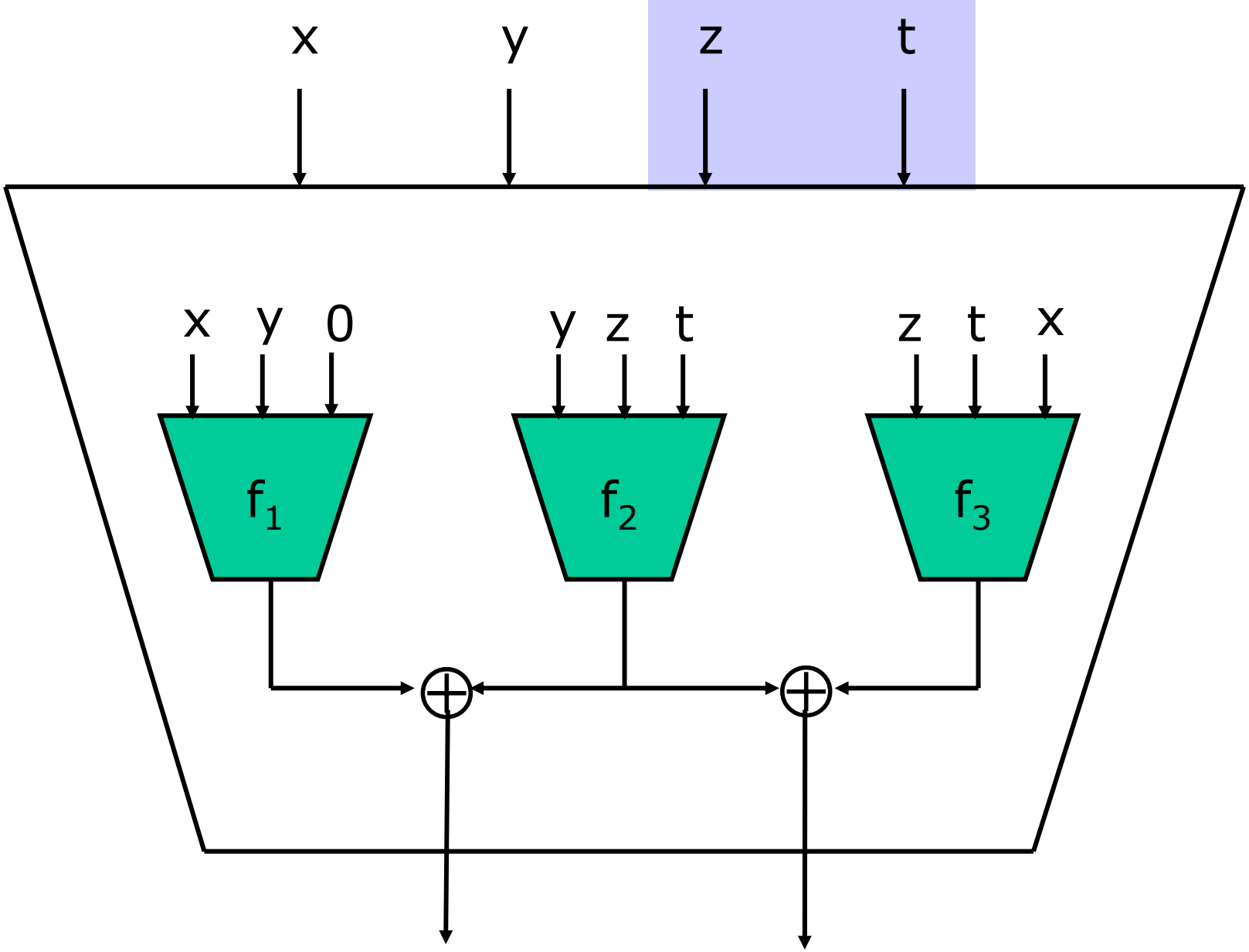
- We showed that the number of queries needed to get a collision is $\Omega(2^{2n/3})$.
- And we showed there exist an attack which makes $O(2^{2n/3})$ queries to get a collision on F.
- So the security bound is tight.

In the security proof

- We do not use the fact that $|x|=|y|=|z|=n$.
- Thus, if we have



i=1,2,3



A double length compression function [rate: 2/3]

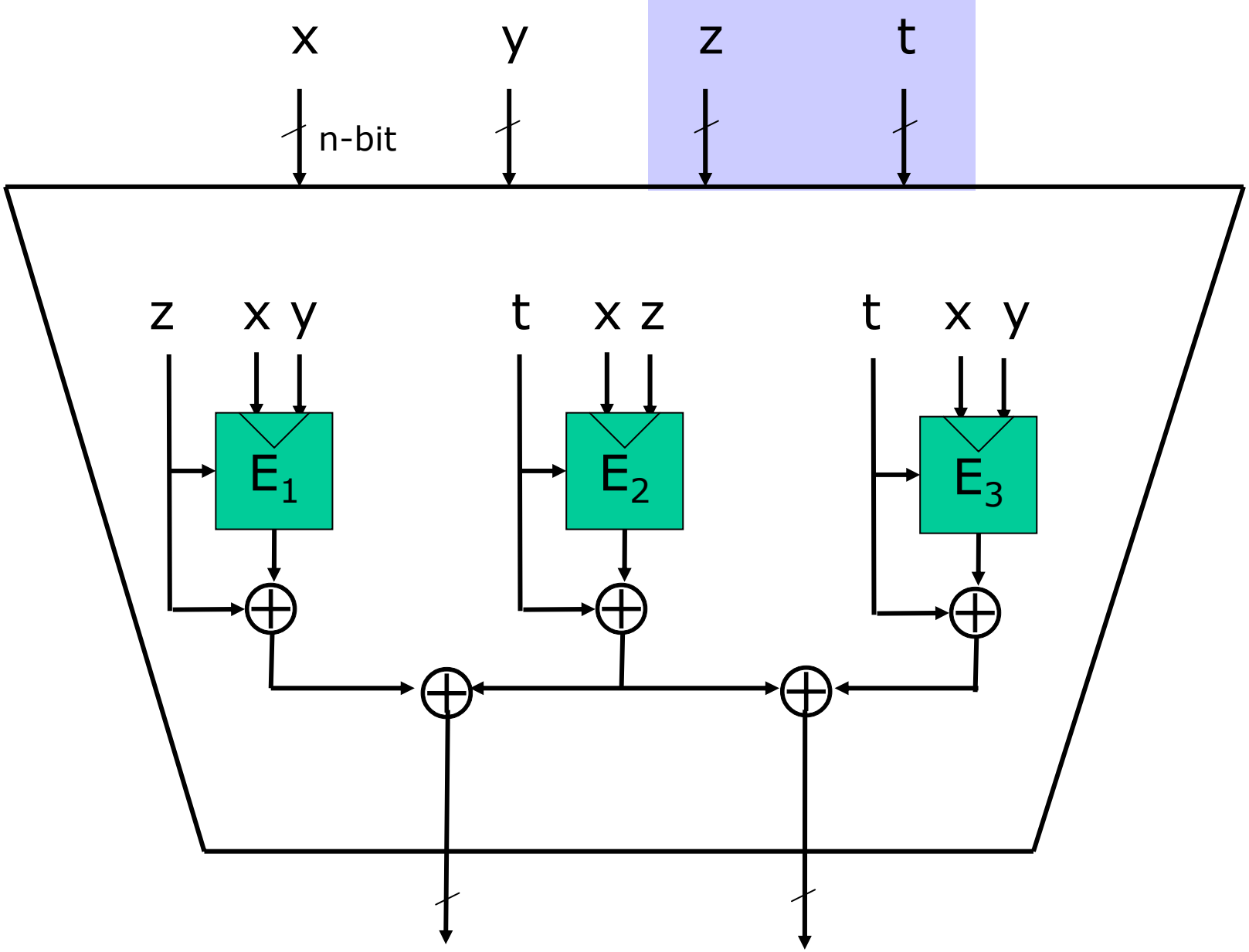
Security

- Then we have same security level as in the previous one.
  - The proof for that is exactly same with the previous proof.

Using the above method

- We can define **a block cipher based double length compression function**.

- We use the block cipher which has **2n**-bit key size and **n**-bit plaintext and ciphertext size.



A block cipher based double length compression function

Adversary : Black-box model

- Adversary can ask both E_i and E_i^{-1} query (i=1,2,3).

  - if he ask (k,x) to oracle E_i, he will get $E_k(x) = y$

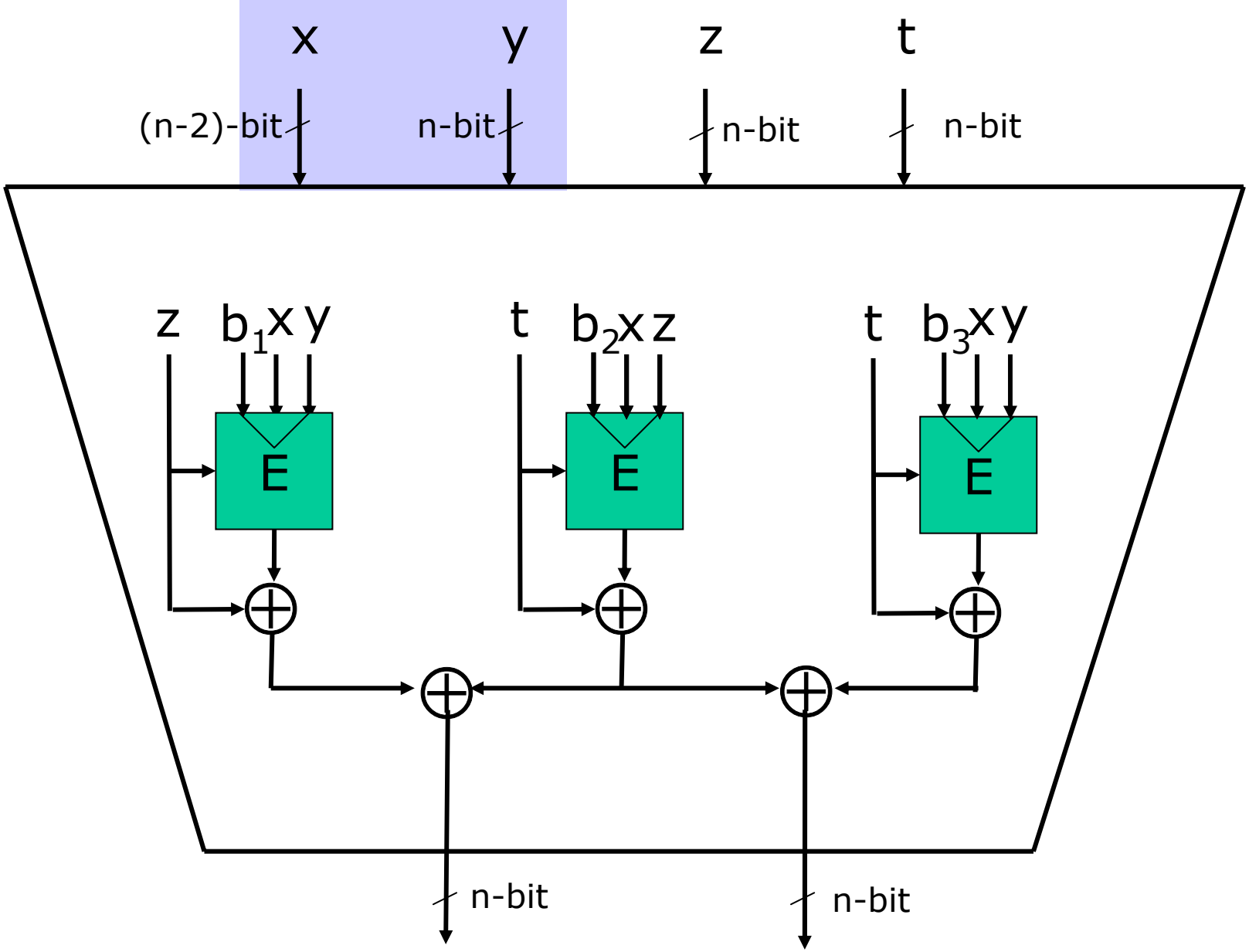
  - if he ask (k,y) to oracle E_i^{-1}, he will get $E^{-1}_k(y) = x$.

Security

- We showed that the number of queries needed to get a collision is $\Omega(2^{2n/3})$.
- We showed a very natural attack which makes $O(2^{2n/3})$ queries to get a collision on F.
- So the security bound is tight.

To use one block cipher

- In order to use only one block cipher, we can use the idea which can be found in the design of MDC-2.



A block cipher based double length compression function

Conclusion

- We proposed a double length compression function which can uses three parallel computations of a compression function or a double key block cipher.

Conclusion

- Although the security is not maximum possible (i.e. there is a better attack than birthday attack), the lower bound of the number of queries is $\Omega(2^{2n/3})$.

- Thus, it has better security than a most secure single length compression function.

Conclusion

- The block cipher based construction is more efficient than the construction (1/2-rate) given in ICISC'04.
  (But, the construction of ICISC'04 is optimal.)

Conclusion

- One can try to design an efficient (if possible, rate-1) double block length hash function which is maximally secure against collision attack even if the underlying compression function is not secure.

Thank you.