Public-Key Identification Schemes based on Multivariate Quadratic Polynomials

Koichi Sakumoto, Taizo Shirai, Harunaga Hiwatari
from Tokyo, Japan

Sony Corporation

@CRYPTO2011

Motivation

- Finding a new alternative to current standard schemes (e.g., RSA) for public-key identification and digital signature

Especially, we would like to provide an alternative based on a problem other than Factoring or DL

Prior works are based on

- ■ Permuted Kernel problem [Shamir '89]
- ■ Syndrome Decoding problem [Stern '93]
- ■ Lattice problem [Micciancio and Vadhan '03]
- ■ …

➡ We focus on an MQ problem

What is an MQ problem?

- Solving a Multivariate Quadratic equation system over a finite field

Given: coefficient a_{lij}, b_{li}, y_l
Find: a solution $(x_1,\cdots,x_n)$

$$F(x_1,\cdots,x_n) = \begin{cases} \Sigma_{ij}\, a_{1ij}\, x_i x_j + \Sigma_i\, b_{1i}\, x_i = y_1 \\ \vdots \\ \Sigma_{ij}\, a_{mij}\, x_i x_j + \Sigma_i\, b_{mi}\, x_i = y_m \end{cases}$$

"MQ function"

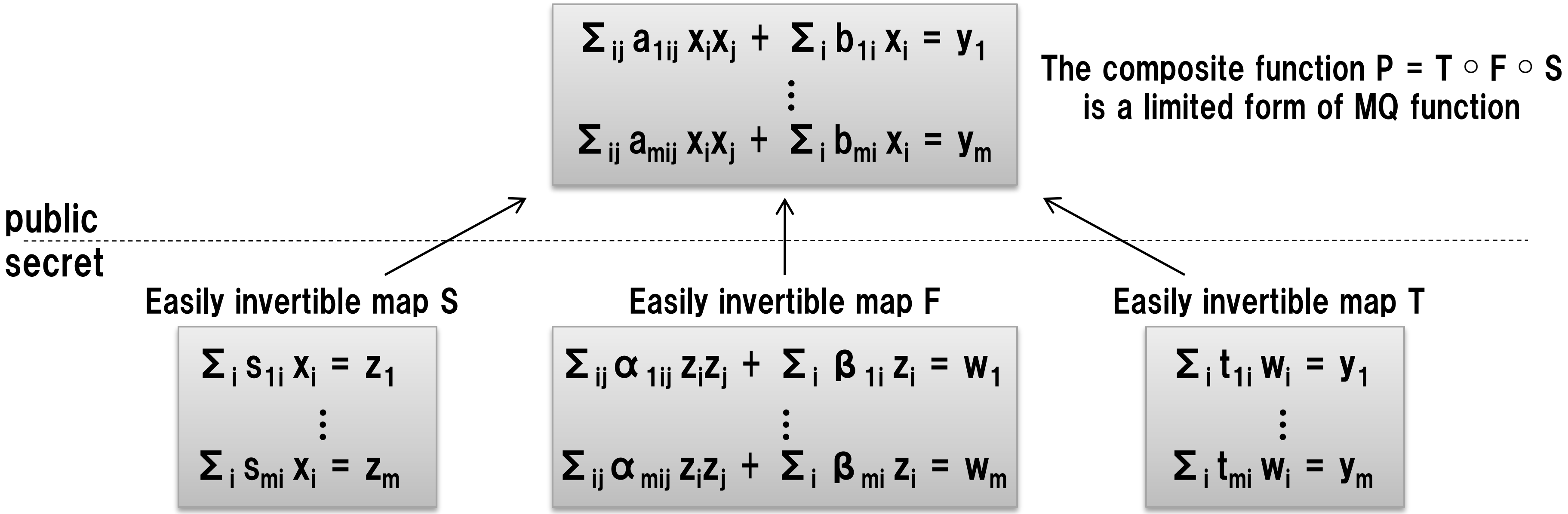
Advantage

- The MQ function can be efficiently implemented
- The MQ function can be used as a one-way function with very short output (e.g., 80 bits)
  - The intractability of a random instance has been well examined
- Associated decision version of the MQ problem is NP-complete
  - There is no known polynomial-time quantum algorithm to solve it

Multivariate Public Key Cryptography (MPKC) uses this form of functions.
But, many existing schemes of MPKC have been already shown to be insecure. Why?

Existing design of Multivariate PKC

- Based on a trapdoor function from composition of easily invertible maps
  - MI scheme [Matsumoto and Imai ‘88] , HFE scheme [Patarin ‘96] , UOV scheme [Kipnis, Patarin, and Goubin ‘99]



$$\begin{gathered}\Sigma_{ij}\, a_{1ij}\, x_i x_j + \Sigma_i\, b_{1i}\, x_i = y_1 \\ \vdots \\ \Sigma_{ij}\, a_{mij}\, x_i x_j + \Sigma_i\, b_{mi}\, x_i = y_m\end{gathered}$$

The composite function $P = T \circ F \circ S$ is a limited form of MQ function

public

secret

Easily invertible map S

$$\begin{gathered}\Sigma_i\, s_{1i}\, x_i = z_1 \\ \vdots \\ \Sigma_i\, s_{mi}\, x_i = z_m\end{gathered}$$

Easily invertible map F

$$\begin{gathered}\Sigma_{ij}\, \alpha_{1ij}\, z_i z_j + \Sigma_i\, \beta_{1i}\, z_i = w_1 \\ \vdots \\ \Sigma_{ij}\, \alpha_{mij}\, z_i z_j + \Sigma_i\, \beta_{mi}\, z_i = w_m\end{gathered}$$

Easily invertible map T

$$\begin{gathered}\Sigma_i\, t_{1i}\, w_i = y_1 \\ \vdots \\ \Sigma_i\, t_{mi}\, w_i = y_m\end{gathered}$$

- The key recovery problem is not an MQ problem, but another problem whose intractability is still controversial
  - The problem is called Isomorphism of Polynomials (IP) problem

In fact, some schemes of MPKC have been already shown to be insecure

Our design

- Based on a zero knowledge argument of knowledge for the MQ problem
  - Especially, a non-trivial and efficient construction by using our original technique

Note: It uses not a composite function, but a random instance of MQ function

System parameter: coefficient a_{lij}, b_{li}
Secret key: input $(x_1,\cdots,x_n)$
Public key: output $(y_1,\cdots,y_n)$

$$\sum_{ij} a_{1ij} x_i x_j + \sum_i b_{1i} x_i = y_1$$

public key

$$\vdots$$

$$\sum_{ij} a_{mij} x_i x_j + \sum_i b_{mi} x_i = y_m$$

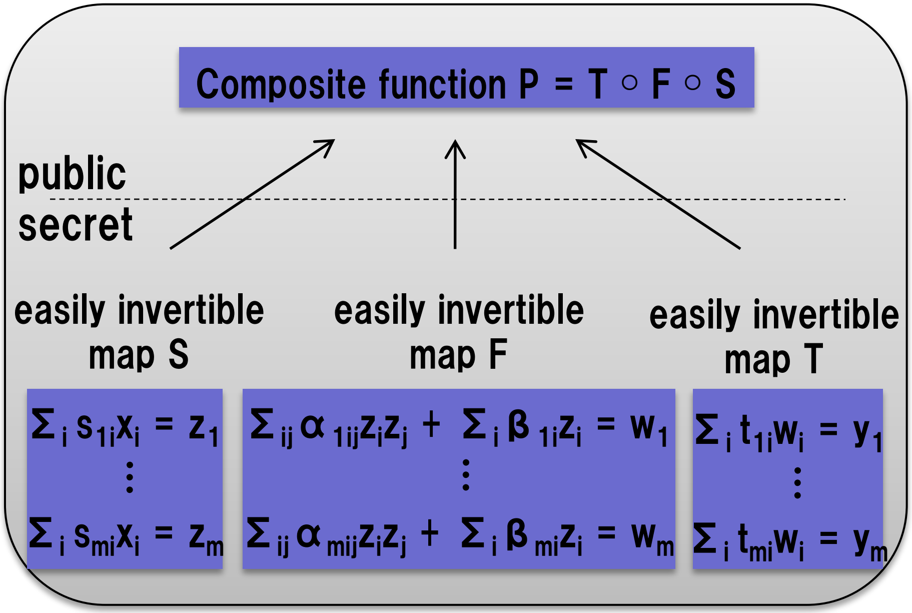
commonly used by all users

Advantage

- The key recovery problem is an MQ problem
  - The security of our scheme can be reduced into the intractability of the MQ problem
- The size of a public key is very small (e.g., 80 bits)

Summary of introduction

- MQ problem is intractable and promising
- We introduce a different design than existing MPKC



Composite function $P = T \circ F \circ S$

public

secret

easily invertible map S

$\Sigma_i s_{1i}x_i = z_1$
$\vdots$
$\Sigma_i s_{mi}x_i = z_m$

easily invertible map F

$\Sigma_{ij} \alpha_{1ij}z_iz_j + \Sigma_i \beta_{1i}z_i = w_1$
$\vdots$
$\Sigma_{ij} \alpha_{mij}z_iz_j + \Sigma_i \beta_{mi}z_i = w_m$

easily invertible map T

$\Sigma_i t_{1i}w_i = y_1$
$\vdots$
$\Sigma_i t_{mi}w_i = y_m$

System parameter: coefficient a_{lij}, b_{li}
Secret key: input $(x_1,\cdots,x_n)$
Public key: output $(y_1,\cdots,y_n)$

$$\Sigma_{ij}\, a_{1ij}\, x_ix_j + \Sigma_i\, b_{1i}\, x_i = y_1$$
$$\vdots$$
$$\Sigma_{ij}\, a_{mij}\, x_ix_j + \Sigma_i\, b_{mi}\, x_i = y_m$$

(a random instance of MQ function)

Existing design of MPKC

Based on a trapdoor function from composition of easily invertible maps

Our design

Based on a zero knowledge argument of knowledge for the MQ problem

Outline

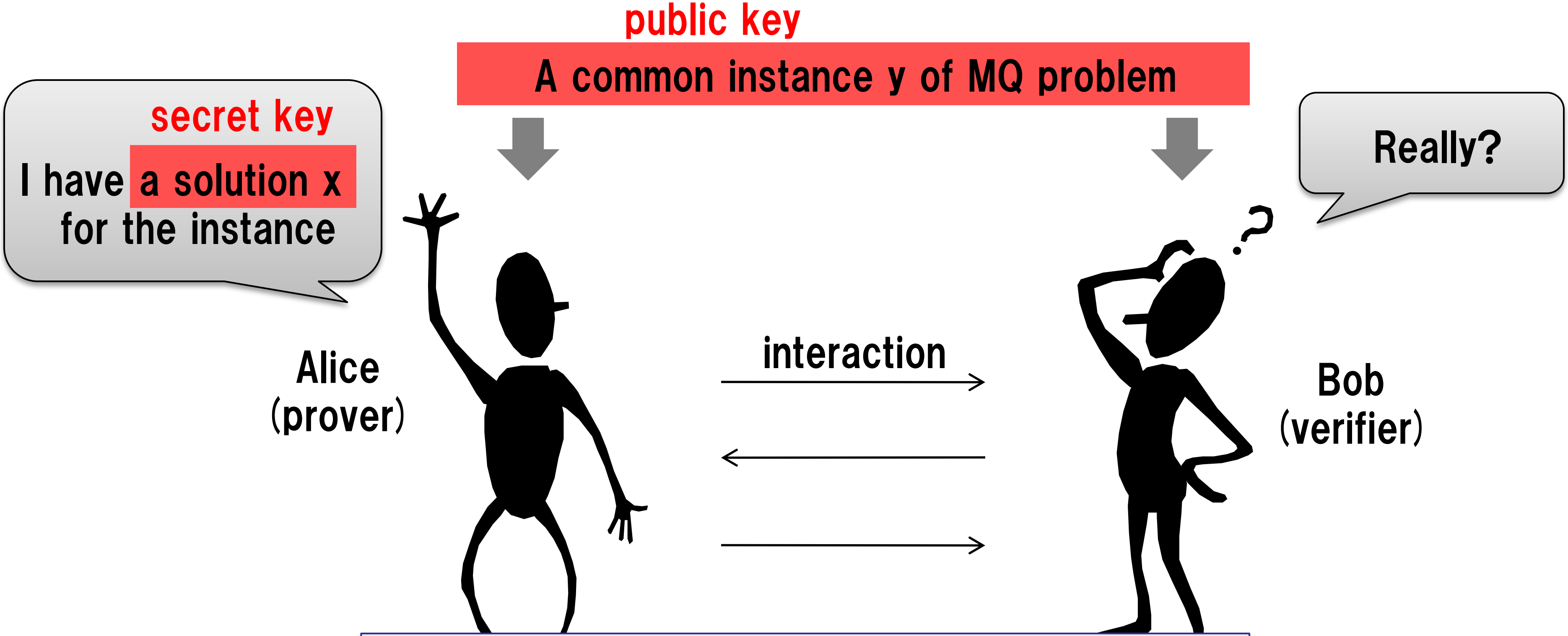
- Introduction
  - Motivation
  - What is an MQ problem
  - Existing design of MPKC
  - Our design
- New technique and construction
  - Zero knowledge argument of knowledge
  - Cut and Choose
  - New technique using the polar form of MQ function
  - Basic protocol
  - Public-key identification scheme
  - Efficiency
- Summary

Outline

- Introduction
  - Motivation
  - What is an MQ problem
  - Existing design of MPKC
  - Our design
- New technique and construction
  - Zero knowledge argument of knowledge
  - Cut and Choose
  - New technique using the polar form of MQ function
  - Basic protocol
  - Public-key identification scheme
  - Efficiency
- Summary

Zero knowledge argument of knowledge

- Alice (Prover) asserts that she has a solution of the MQ problem
- Bob (Verifier) checks whether the assertion is true or not through interaction with Alice



public key

A common instance y of MQ problem

secret key

I have a solution x for the instance

Really?

Alice (prover)

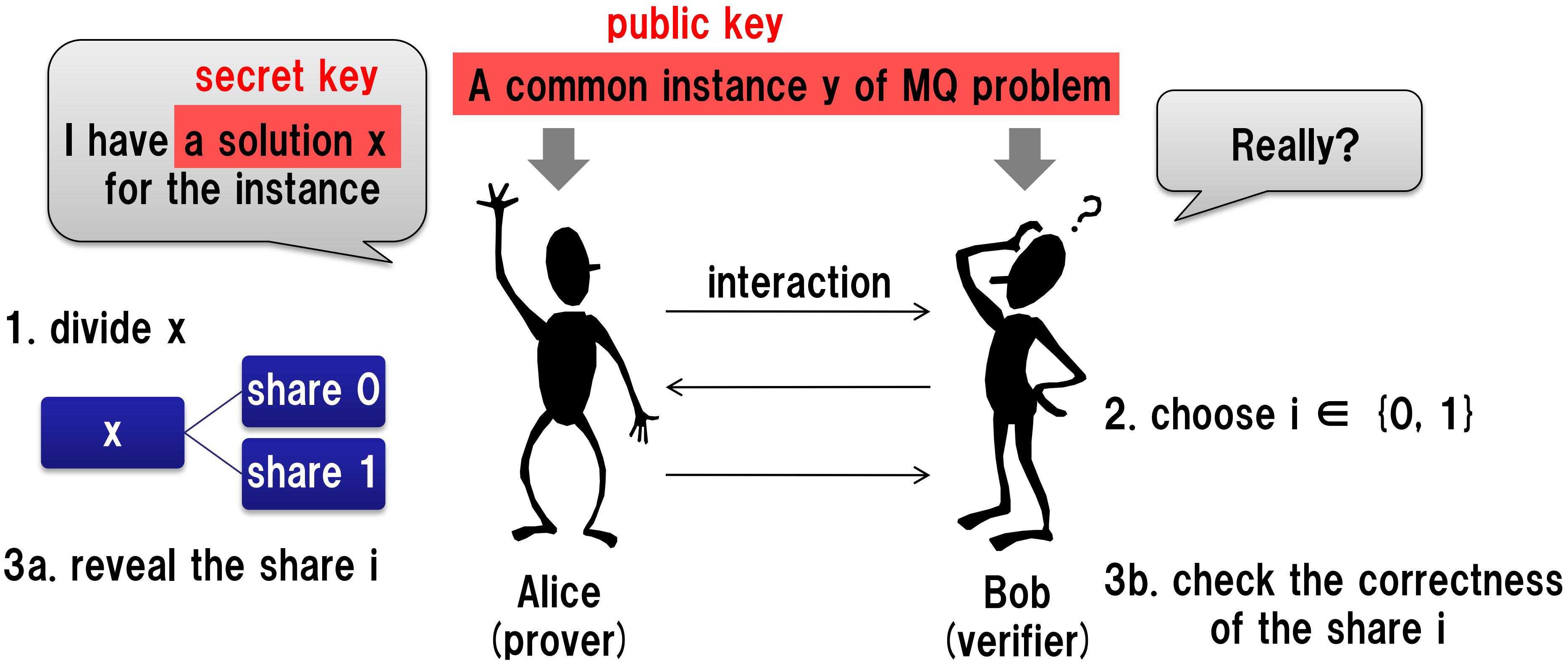
interaction

Bob (verifier)

Security can be derived from two properties, “zero knowledge” and “argument of knowledge”

Cut-and-Choose approach

1. Alice (prover) divides her secret into shares
2. Bob (verifier) chooses which share he checks
3. She proves the correctness of the chosen share without revealing her secret itself



public key

A common instance y of MQ problem

secret key

I have a solution x for the instance

Really?

1. divide x

x

share 0

share 1

interaction

2. choose i ∈ {0, 1}

3a. reveal the share i

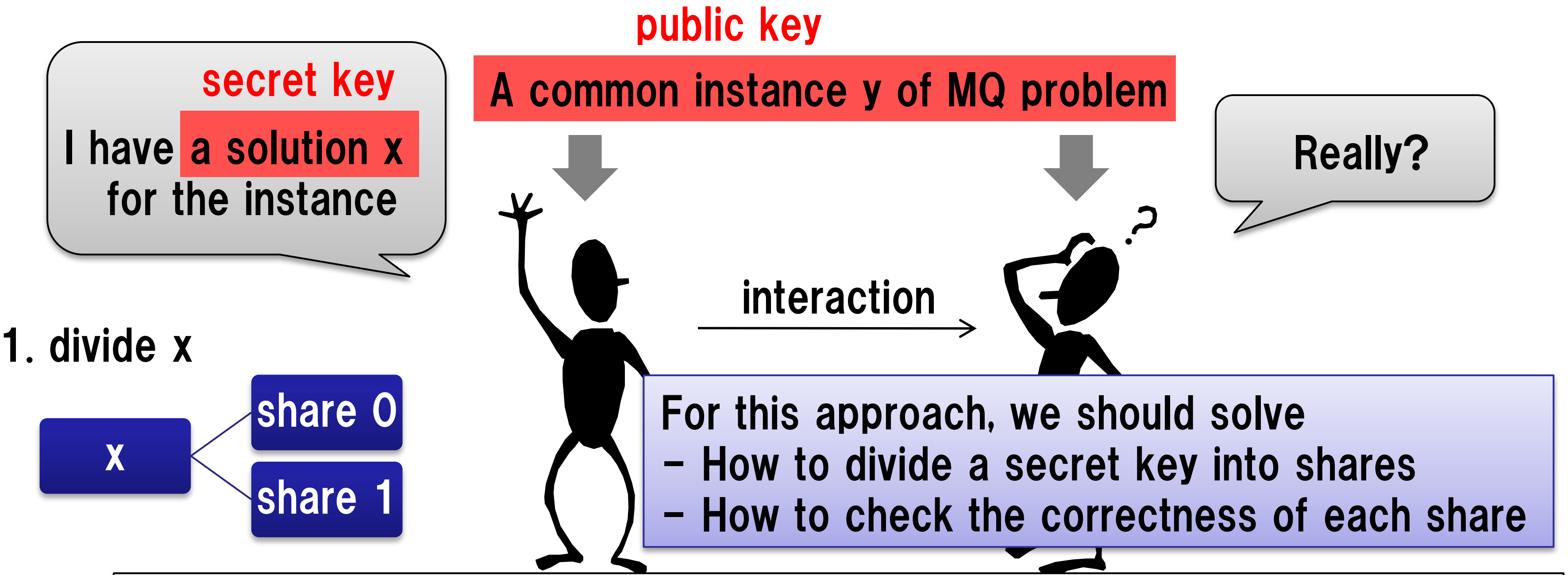
3b. check the correctness of the share i

Alice (prover)

Bob (verifier)

Cut-and-Choose approach

1. Alice (prover) divides her secret into shares
2. Bob (verifier) chooses which share he checks
3. She proves the correctness of the chosen share without revealing her secret itself


public key
A common instance y of MQ problem
secret key
I have a solution x for the instance
Really?
interaction
1. divide x
x
share 0
share 1

For this approach, we should solve
- How to divide a secret key into shares
- How to check the correctness of each share

3a. re

ness

A property of homomorphism is useful (e.g., modular exponentiation)
If a secret key x is divided into r_0 + r_1,
then a public key g^x is correspondingly divided into $g^{r_0} * g^{r_1}$

New Cut-and-Choose technique

For an MQ function F, consider a situation where

- Secret key: x
- Public key: $y = F(x)$

A useful property

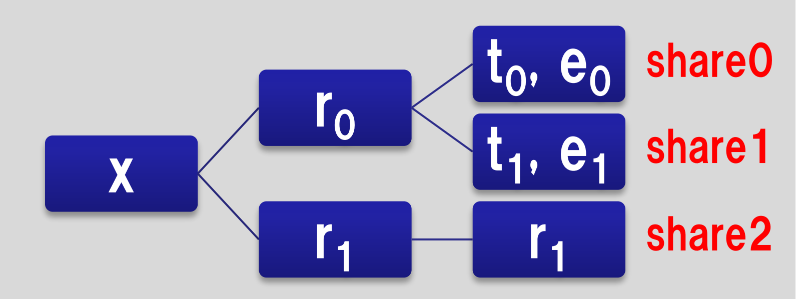
The associated polar form $G(x,y)$ of $F(x)$

$$G(x,y) = F(x+y) - F(x) - F(y)$$

is a bilinear function

By using the useful property, divide a secret key into three shares:

- First, divide $x = r_0 + r_1$
  - Consequently, y is divided $y = F(r_0+r_1) = G(r_0,r_1) + F(r_0) + F(r_1)$
- Second, further divide $r_0 = t_0 + t_1$ and $F(r_0) = e_0 + e_1$
  - Consequently, $y = \underbrace{G(t_0,r_1) + e_0}_{\text{share0 and share2}} + \underbrace{F(r_1) + G(t_1,r_1) + e_1}_{\text{share1 and share2}}$



Note

No information on the secret key x can be extracted from only two out of the three shares

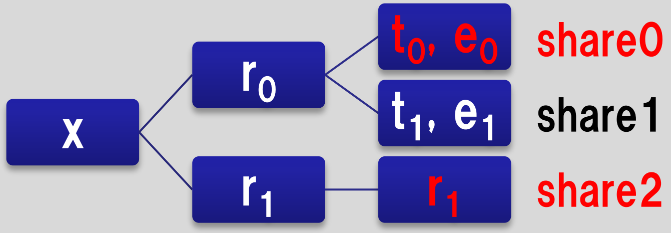
Our basic protocol

Secret key: x

Public key: y = F(x)

Secret key x → Alice (prover) | Bob (verifier)

Divide x into three shares



x → r_0, r_1

r_0 → t_0, e_0 share0

r_0 → t_1, e_1 share1

r_1 → r_1 share2

Randomly pick r_0, t_0, e_0

$r_1 \leftarrow x - r_0$, $t_1 \leftarrow r_0 - t_0$,

$e_1 \leftarrow F(r_0) - e_0$

$c_0 \leftarrow \mathrm{Com}(r_1, G(t_0, r_1) + e_0)$

$c_1 \leftarrow \mathrm{Com}(t_0, e_0)$

$c_2 \leftarrow \mathrm{Com}(t_1, e_1)$

Commit these values

(c_0, c_1, c_2) →

Randomly pick Ch ∈ {0,1,2}

← Ch

If Ch=0 then $\sigma = (r_0, t_1, e_1)$

If Ch=1 then $\sigma = (r_1, t_1, e_1)$

If Ch=2 then $\sigma = (r_1, t_0, e_0)$

σ →

share0 and share1

If Ch=0 then check

$c_1 = \mathrm{Com}(r_0 - t_1, F(r_0) - e_1)$, $c_2 = \mathrm{Com}(t_1, e_1)$

share1 and share2

If Ch=1 then check

$c_0 = \mathrm{Com}(r_1, y - F(r_1) - G(t_1, r_1) - e_1)$

$c_2 = \mathrm{Com}(t_1, e_1)$

share0 and share2

If Ch=2 then check

$c_0 = \mathrm{Com}(r_1, G(t_0, r_1) + e_0)$, $c_1 = \mathrm{Com}(t_0, e_0)$

Our basic protocol

Secret key x → Alice (prover)

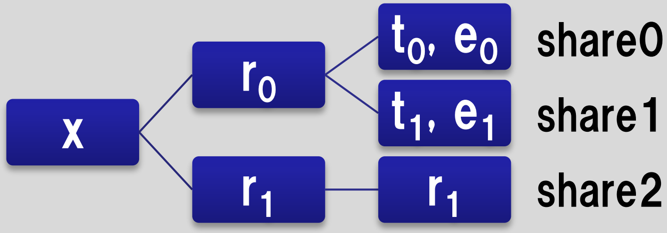
Bob (verifier)

Secret key: x
Public key: y = F (x)

Randomly pick r_0, t_0, e_0
$r_1 \leftarrow x - r_0$, $t_1 \leftarrow r_0 - t_0$,
$e_1 \leftarrow F(r_0) - e_0$
$c_0 \leftarrow \mathrm{Com}(r_1, G(t_0, r_1) + e_0)$
$c_1 \leftarrow \mathrm{Com}(t_0, e_0)$
$c_2 \leftarrow \mathrm{Com}(t_1, e_1)$

Commit these values

Divide x into three shares



$\xrightarrow{(c_0, c_1, c_2)}$

Randomly pick Ch ∈ {0,1,2}

$\xleftarrow{\mathrm{Ch}}$

If Ch=0 then $\sigma = (r_0, t_1, e_1)$
If Ch=1 then $\sigma = (r_1, t_1, e_1)$
If Ch=2 then $\sigma = (r_1, t_0, e_0)$

$\xrightarrow{\sigma}$

share0 and share1

If Ch=0 then check
$c_1 = \mathrm{Com}(r_0 - t_1, F(r_0) - e_1)$, $c_2 = \mathrm{Com}(t_1, e_1)$

share1 and share2

If Ch=1 then check

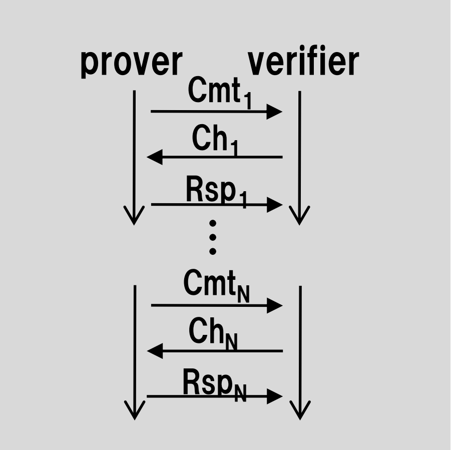
$e_0)$

Theorem

- This protocol is statistically zero knowledge when Com is statistically hiding.
- This protocol is argument of knowledge for the MQ problem
with knowledge error 2/3 when Com is computationally binding.

Public-key identification schemes

Sequential Composition


prover verifier

Cmt_1

Ch_1

Rsp_1

⋮

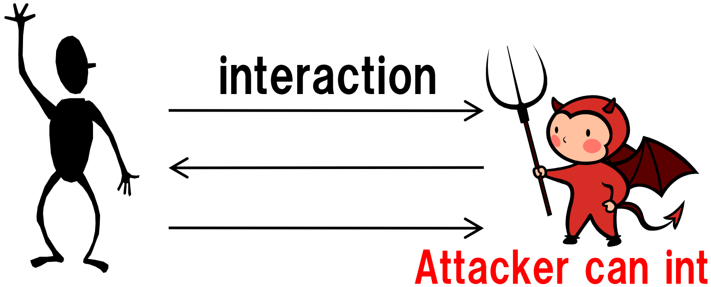
Cmt_N

Ch_N

Rsp_N

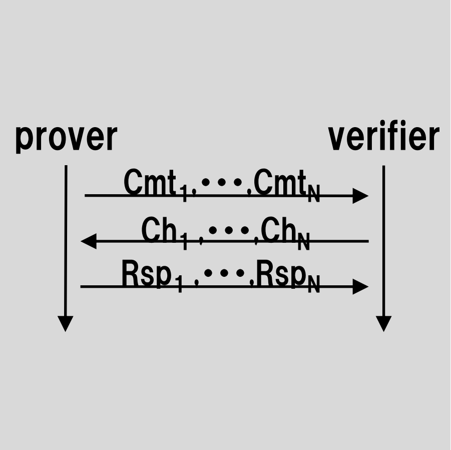
Achieve the security

against **active attack**



Attacker can interact with an honest prover

Parallel Composition


prover verifier

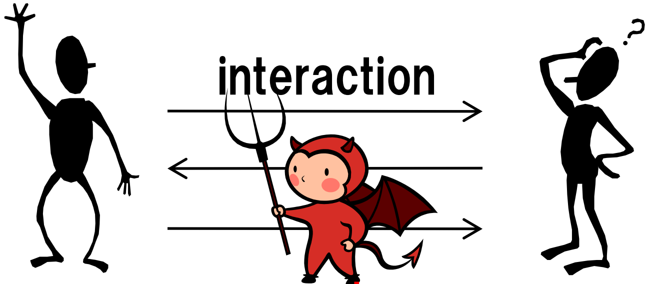
$\mathrm{Cmt}_1,\cdots,\mathrm{Cmt}_N$

$\mathrm{Ch}_1,\cdots,\mathrm{Ch}_N$

$\mathrm{Rsp}_1,\cdots,\mathrm{Rsp}_N$

Achieve the security

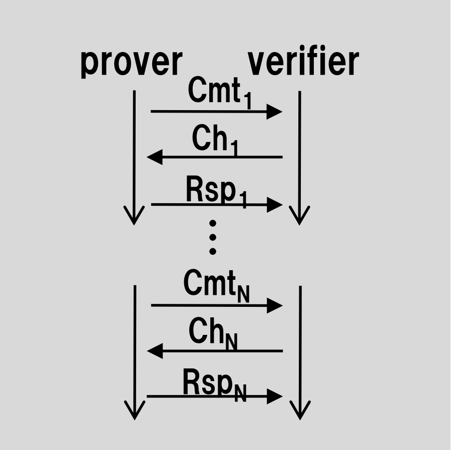
against **passive attack**



eavesdropper

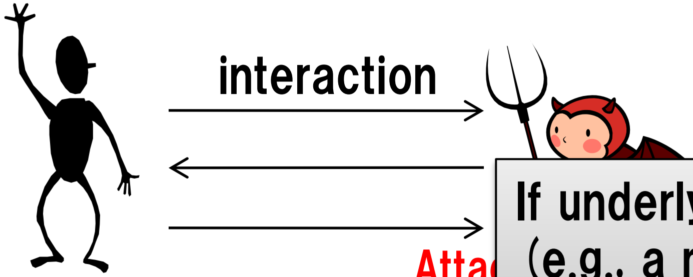
Public-key identification schemes

Sequential Composition



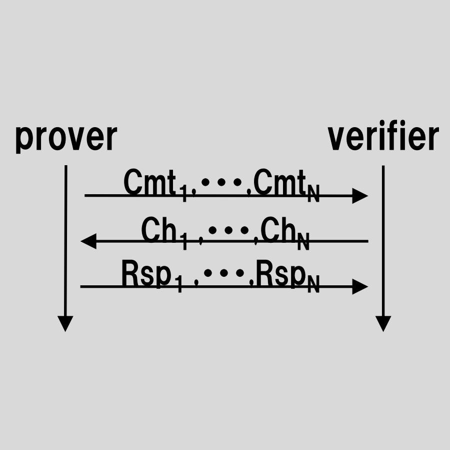
Achieve the security

against **active attack**



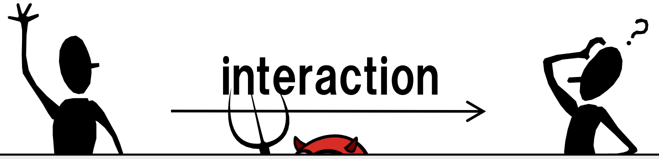
Attac
with

Parallel Composition


prover
verifier
$\mathrm{Cmt}_1, \cdots, \mathrm{Cmt}_N$
$\mathrm{Ch}_1, \cdots, \mathrm{Ch}_N$
$\mathrm{Rsp}_1, \cdots, \mathrm{Rsp}_N$

Achieve the security

against **passive attack**



If underlying MQ function is substantially compressing (e.g., a map from 160 bits to 80 bits), the parallel version also achieves the security against **active attack**

Efficiency

- Comparison with public-key identification schemes based on another problem whose associated decisional version is NP-complete
  - The schemes from 3-pass zero knowledge argument of knowledge

Problem	SD [Stern '93]	CLE [Stern '94]	PP [Pointcheval '95]	MQ [Ours]
Public key size for 80-bit security	350 bit	288 bit	245 bit	80 bit
Communication data size	7.5 KByte	5.7 KByte	12.6 KByte	3.7 KByte
Arithmetic operations	2^{24} / F_2	2^{16} / F_{257}	2^{22} / F_{127}	2^{26} / F_2
Random permutation	S_{700}	S_{24}	S_{161},S_{177}	Not required

  - In the case that the protocol is repeated until the impersonation probability is less than 2^{-30} (< 1/one billion)

[Stern '93] "A New Identification Scheme Based on Syndrome Decoding", J. Stern.
[Stern '94] "Designing Identification Schemes with Keys of Short Size", J. Stern.
[Pointcheval '95] "New Identification Scheme Based on the Perceptrons Problem", D. Pointcheval.

Summary

- We proposed public-key identification schemes based on an MQ problem
  - New design: different from existing MPKC
    - Based on a zero knowledge argument of knowledge for the MQ problem
  - Advantage: the security and the public key size
    - The security can be reduced into the intractability of a random instance of MQ problem
    - The size of a public key is very small（e.g., 80 bits)
- Another application
  - Digital signature scheme

Thank you for your attention!