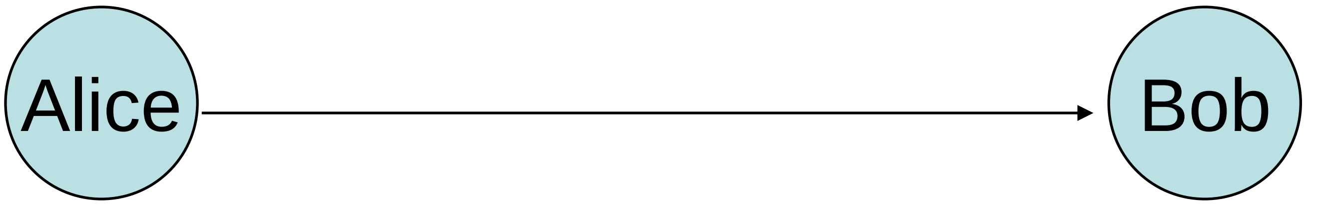
Truly Efficient 2-Round Perfectly Secure Message Transmission Scheme

Kaoru Kurosawa

Kazuhiro Suzuki

(Ibaraki University, Japan)

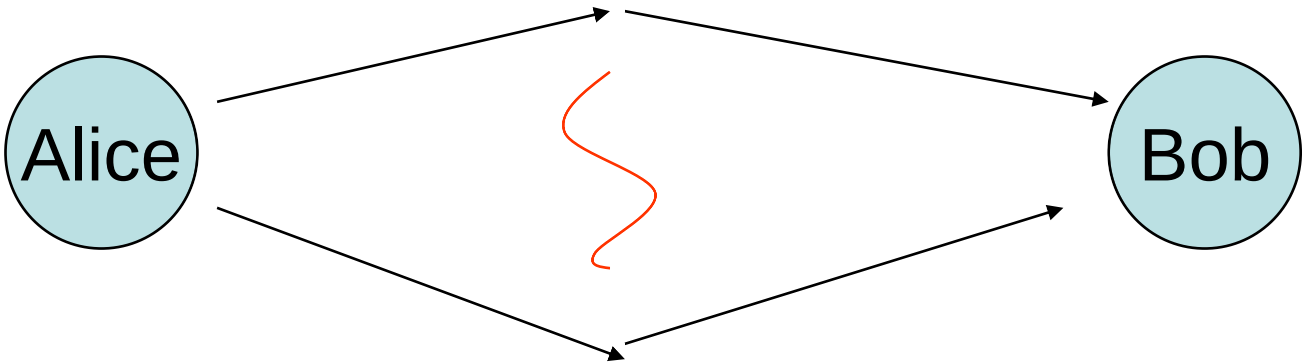
Usual Model of Encryption



- Single line between Alice and Bob.
- Alice and Bob share a key.
- Enemy can fully corrupt the channel.

(Observe and modify the ciphertext)

Dolev, Dwork, Waarts and Yung

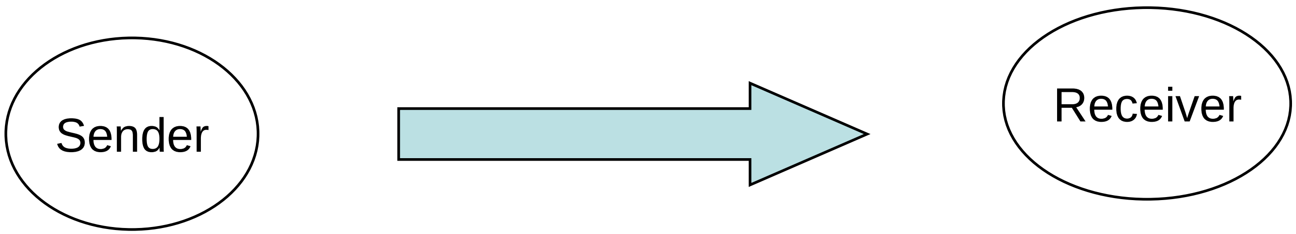


- n-channels between Alice and Bob.
- An infinitely powerful adversary A can corrupt t out of n channels. (Observe and modify)

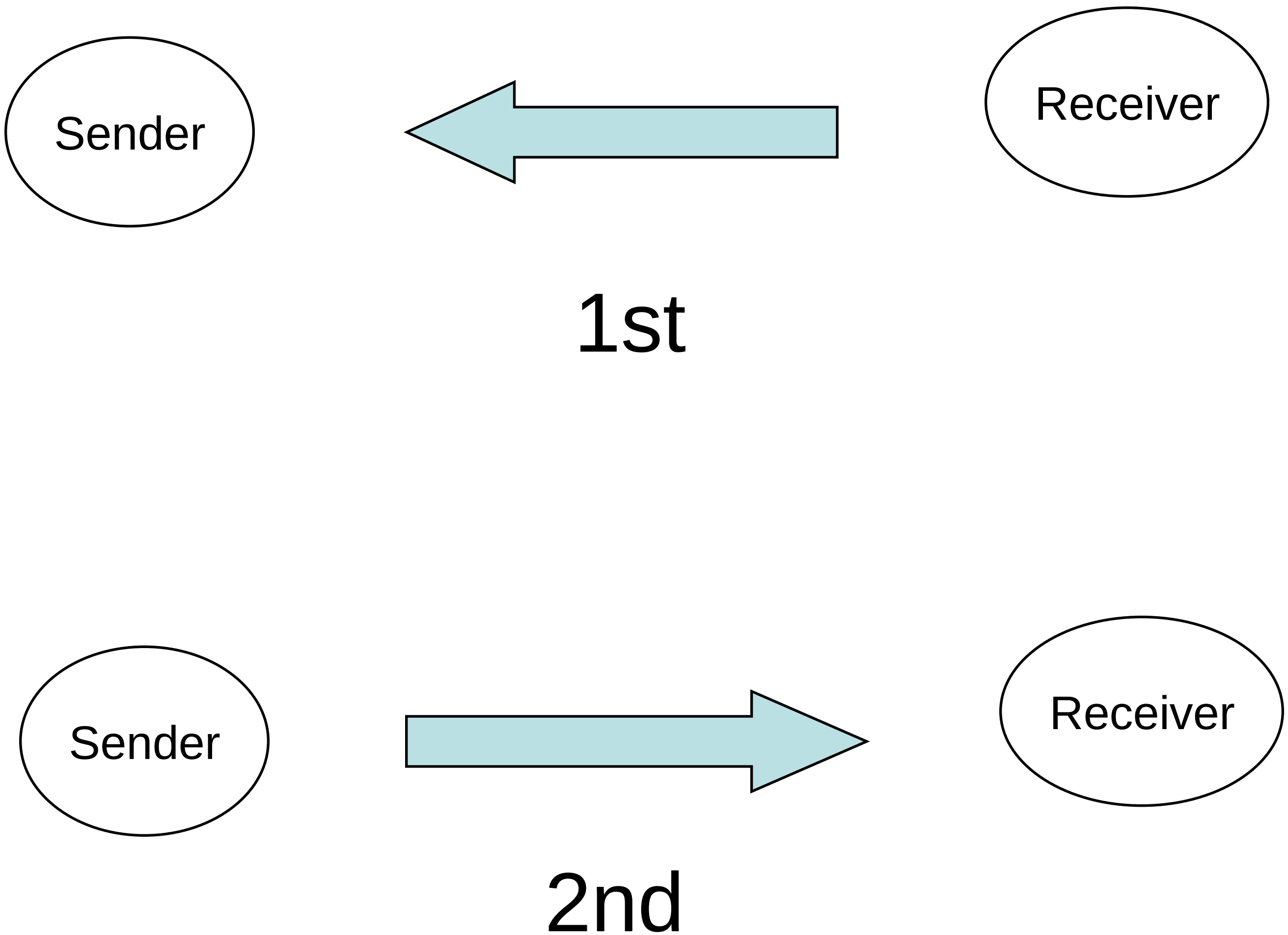
Goal

- Alice wishes to send a secret s to Bob
- in r-rounds
- without sharing any key.

1 Round Protocol



2 Round Protocol



We say that a MT scheme

is perfectly secure if

- (Perfect Privacy)

  Adversary learns no information on s

- (Perfect Reliability)

  Bob can receive s correctly

In what follows, PSMT means

- Perfectly
- Secure
- Message
- Transmission
- Scheme

For 1 round,

- Dolev et al. showed that there exists a 1-round PSMT iff $n \geqq 3t+1$.
- They also showed an efficient 1-round PSMT.

where the adversary can corrupt t out of n channels.

For 2 rounds,

- It is known that there exists a 2-round PSMT iff $n \geqq 2t+1$.

- However, it is very difficult to construct an efficient scheme for $n=2t+1$.

For n=2t+1,

- Dolev et al. showed a 3-round PSMT such that the transmission rate is $O(n^5)$,
- where the transmission rate is defined as

$$\frac{\text{the total number of bits transmitted}}{\text{the size of the secrets}}$$

Sayeed et al. showed

- a 2-round PSMT such that
  the transmission rate is $O(n^3)$

Srinathan et al. showed that

- n is a lower bound on
  the transmission rate of 2-round PSMT
  with n=2t+1.

At CRYPTO 2006,

- Agarwal, Cramer and de Haan
  showed a 2-round PSMT such that
  the transmission rate is O(n) .

- However,
  the computational cost is exponential.

Agarwal, Cramer and de Haan

- left it as an open problem to construct a 2-round PSMT for $n=2t+1$ such that
- not only

  the transmission rate is $O(n)$
- but also

  the computational cost is poly(n).

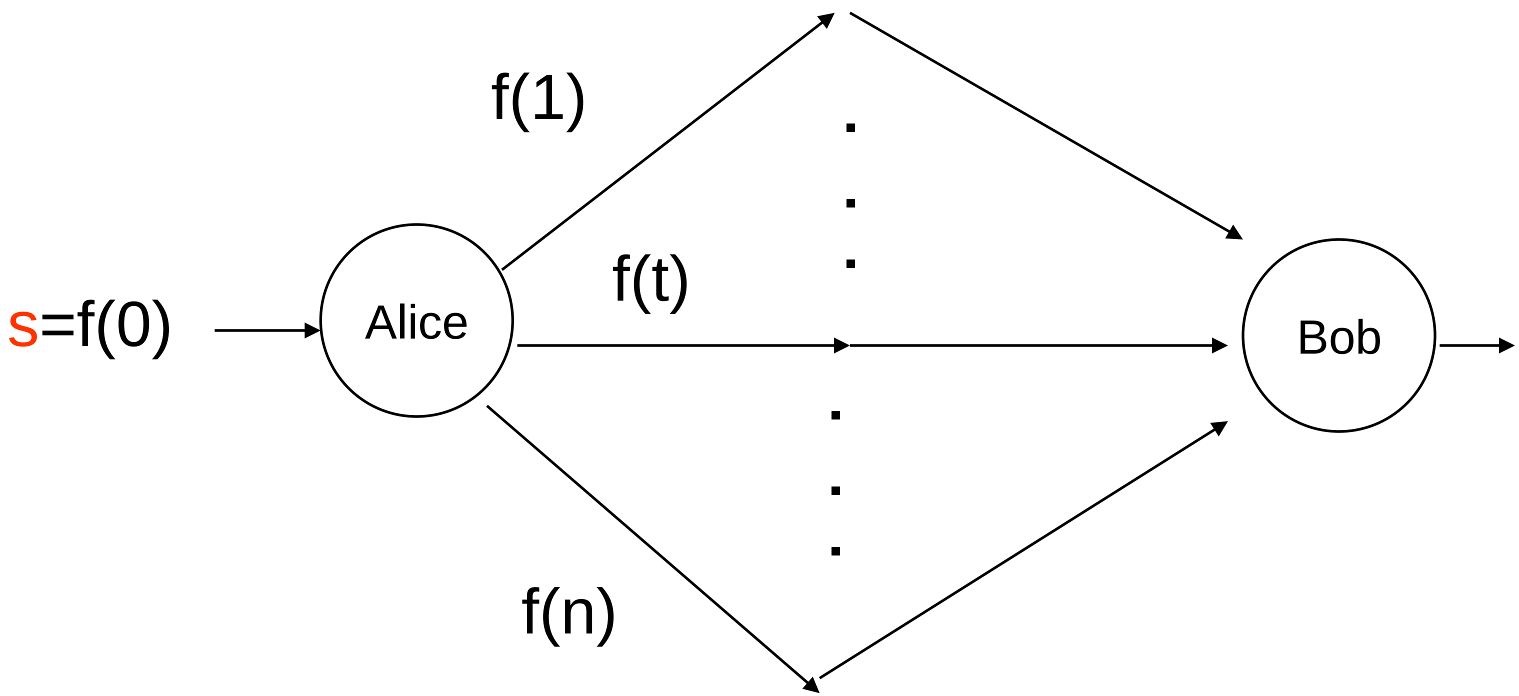
In This Paper,

- We solve this open problem.

2-round PSMT for n=2t+1

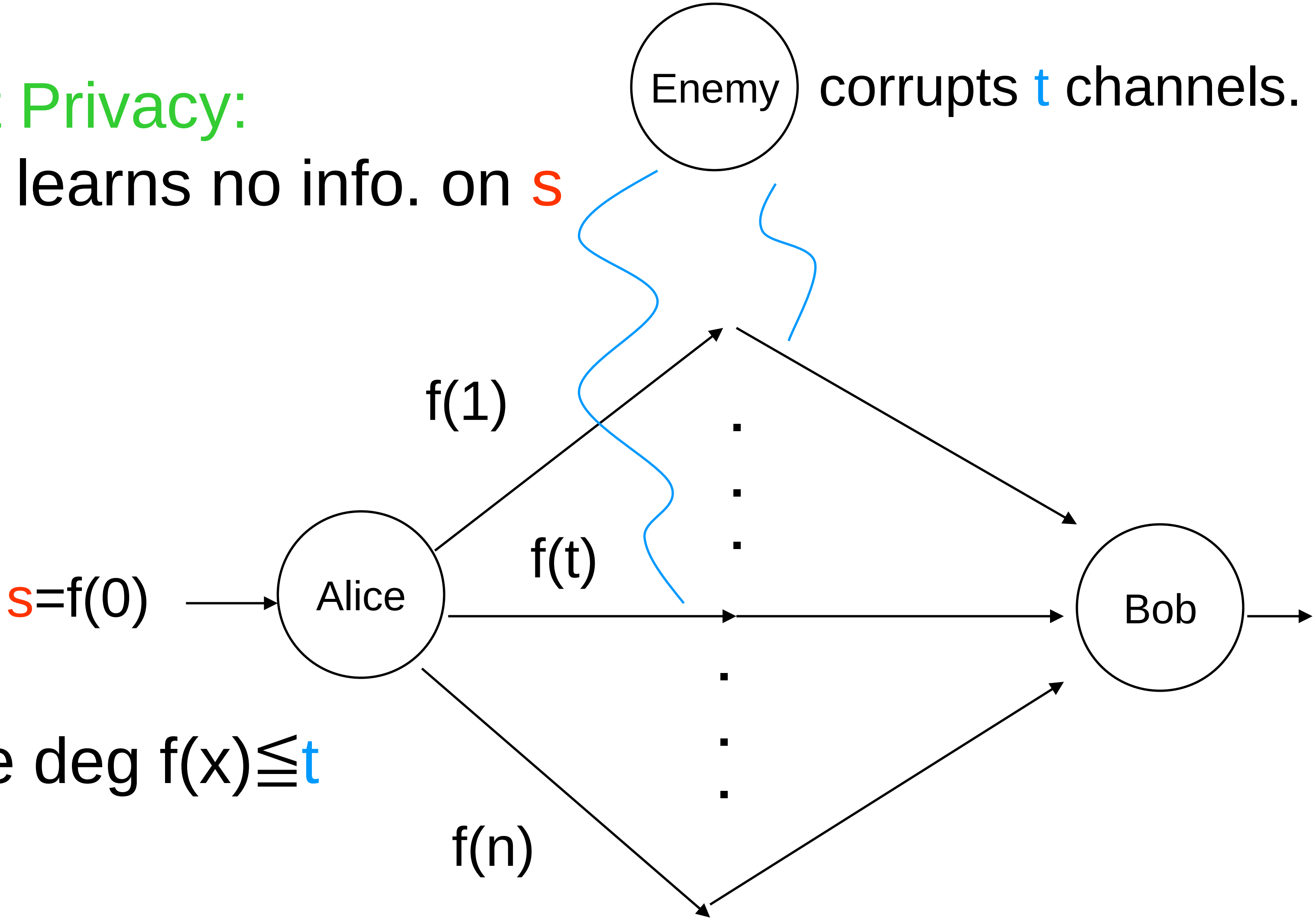
	Trans. rate	Sender's comp.	Receiver's comp.
Agarwal et al.'s schme	O(n)	exponential	exponential
Proposed scheme	O(n)	poly(n)	poly(n)

Consider a MT as follows.
Alice chooses a random f(x) such that
deg f(x)≦t and



Perfect Privacy:

Enemy learns no info. on s



because deg $f(x) \leqq t$

Let C be a linear code

- such that a codeword is

  $X=(f(1),\ldots, f(n))$,
- where $f(x)$ is a polynomial

  with $\deg f(x) \leqq t$.

Let C be a linear code

- such that a codeword is
  $X=(f(1),\ldots, f(n))$,
- where with $\deg f(x) \leqq t$.

- Then X has at most t zeros
  because $\deg f(x) \leqq t$.

Let C be a linear code

- such that a codeword is

  $X=(f(1),\ldots, f(n))$,
- where with $\deg f(x) \leqq t$.
- Then X has at most t zeros.
- Hence

  the minimum Hamming weight of C is

  n-t.

Let C be a linear code

- such that a codeword is

  $X=(f(1),\ldots, f(n))$,
- where with $\deg f(x) \leqq t$.
- Then X has at most t zeros.
- Hence

  the minimum Hamming distance of C is

  $d=n-t$.

If $n=3t+1$,

- the minimum Hamming distance of C is

  $d = n - t = (3t+1) - t = 2t+1.$

If $n=3t+1$,

- the minimum Hamming distance of C is

  $d=n - t = (3t+1) - t = 2t+1.$

- Hence the receiver can correct t errors caused by the adversary.

If $n=3t+1$,

- the minimum Hamming distance of C is

  $$d=n-t=(3t+1)-t=2t+1.$$

- Hence the receiver can correct t errors caused by the adversary.
- Thus perfect reliability is also satisfied.

If $n=3t+1$,

- the minimum Hamming distance of C is

  $$d = n - t = (3t+1) - t = 2t+1.$$

- Hence the receiver can correct t errors caused by the adversary.
- Thus perfect reliability is satisfied.
- Therefore

  we can obtain a 1-round PSMT easily.

If $n=2t+1$, however,

- the minimum Hamming distance of C is

  $d = n - t = (2t+1) - t = t+1$

If $n=2t+1$, however,

- the minimum Hamming distance of C is

  $d=n-t=(2t+1)-t= t+1$
- Hence the receiver can only detect t errors, but cannot correct them.

If $n=2t+1$, however,

- the minimum Hamming distance of C is

  $d=n-t=(2t+1)-t=t+1$
- Hence the receiver can only detect t errors, but cannot correct them.
- This is the main reason why the construction of PSMT for $n=2t+1$ is difficult.

What is a difference

- between error correction and PSMT ?

What is a difference

- between error correction and PSMTs ?
- If the sender sends a single codeword,
  then the Enemy causes t errors randomly.

What is a difference

- between error correction and PSMTs ?
- If the sender sends a single codeword, then the Enemy causes t errors randomly.
- Hence there is no difference.

Our Observation

- If the sender sends many codewords

  $X_1, \ldots, X_m,$

  then the errors are not totally random
- because

  the errors always occur

  at the same t (or less) places !

Our Observation

- Suppose that the receiver received

  $Y_1 = X_1 + E_1, \ldots, \; Y_m = X_m + E_m,$
- Let

  $$E = [E_1, \ldots, E_m].$$
- Then

  $$\dim E \leqq t$$

  because the errors always occur at the same t (or less) places !

Suppose that the receiver received

$$Y_i = X_i + E_i$$

$Y = \{Y_1, \ldots, Y_m\}$	$E = [E_1, \ldots, E_m]$.
Pseudo dim k $\Longleftrightarrow$	dim k
Pseudo basis $\{Y_{j1}, \ldots, Y_{jk}\}$ $\Longleftrightarrow$	Basis $\{E_{j1}, \ldots, E_{jk}\}$

Main Contribution

- We introduce a notion of

$$\left\{\begin{array}{l}\text{pseudo-dimension}\\ \text{pseudo-basis,}\end{array}\right.$$

and

- show a poly-time algorithm

which finds them from $Y=\{Y_1, \ldots, Y_m\}$.

Main Contribution

- We introduce a notion of

  $\left\{\begin{array}{l}\text{pseudo-dimension} \\ \text{pseudo-basis, and}\end{array}\right.$

- show a poly-time algorithm which finds them from $Y=\{Y_1, \ldots, Y_m\}$.
- Please see the proceedings for this algorithm.

More Observation

For example,

- $E_1=(1,0, \ldots, 0)$,
- $E_2=(1,1,0, \ldots, 0)$,
- $\ldots$
- $E_t=(1,\ldots,1,0, \ldots, 0)$,

is a basis of E.

More Observation

- $E_1=(1,0, \ldots, 0)$, $\mathrm{NonZero}(E_1)=\{1\}$
- $E_2=(1,1,0, \ldots, 0)$, $\mathrm{NonZero}(E_2)=\{1,2\}$
- …
- $E_t=(1,\ldots,1,0, \ldots, 0)$, $\mathrm{NonZero}(E_t)=\{1, \ldots, t\}$

More Observation

- $E_1=(1,0,\ldots,0)$, NonZero(E_1)={1}
- $E_2=(1,1,0,\ldots,0)$, NonZero(E_2)={1,2}
- …
- $E_t=(1,\ldots,1,0,\ldots,0)$, NonZero($E_t$)={1, …, t}

- Define

  FORGED = U NonZero(E_i)

  basis

More Observation

- $E_1=(1,0, \ldots, 0)$, NonZero(E_1)={1}
- $E_2=(1,1,0, \ldots, 0)$, NonZero(E_2)={1,2}
- …
- $E_t=(1,\ldots,1,0, \ldots, 0)$, NonZero($E_t$)= {1, …, t}

- Define

  FORGED = U NonZero(E_i)

  basis

  = {all forged channels}

In general,

- FORGED = U NonZero(E_i)

  basis

FORGED = {all forged channels}

Rest of This Talk

- Our 3-round PSMT
- Basic 2-round PSMT
- More Efficient 2-round PSMT
- Final 2-round PSMT

Rest of This Talk

- Our 3-round PSMT
- Basic 2-round PSMT
- More Efficient 2-round PSMT
- Final 2-round PSMT

For i=1, …, t+1,

Sender

Random codeword

$X_i = (f_i(1), \ldots, f_i(n))$

$Y_i = X_i + E_i$

Receiver

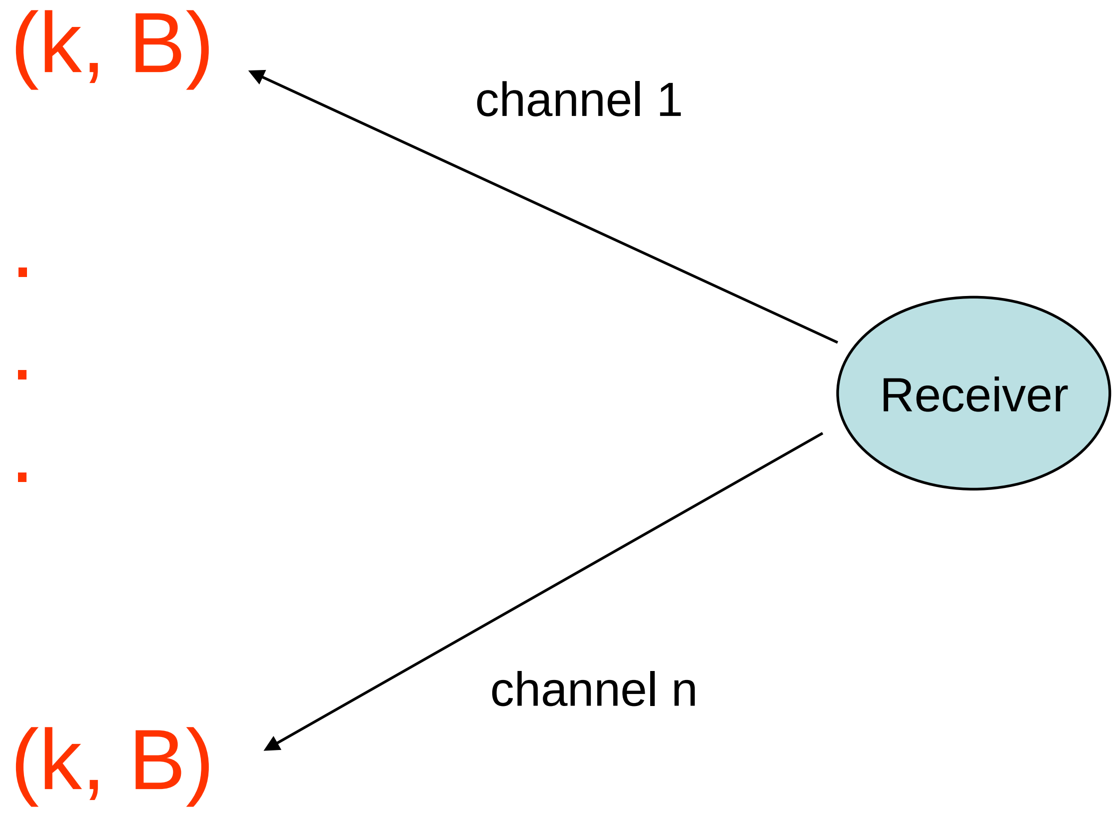
Pseudo-dimension k

Pseudo-basis B

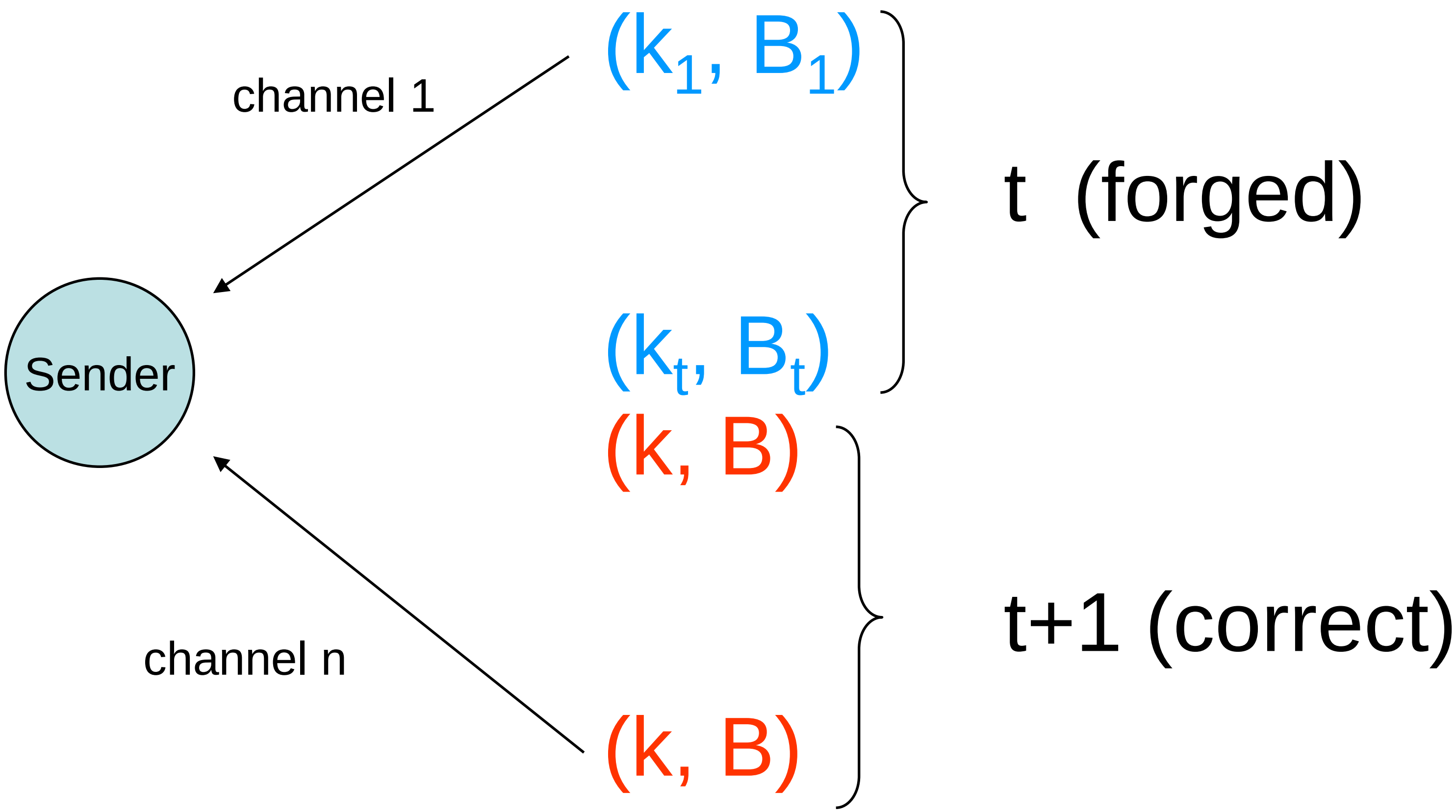
of $\{Y_1, \ldots, Y_{t+1}\}$

Receiver

R Broadcasts (k, B)

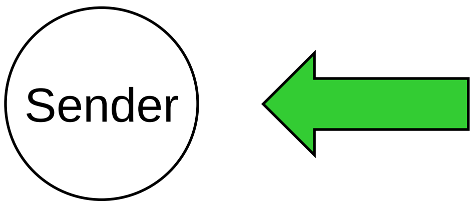


S can receive them correctly by taking the majority vote



because n = 2t + 1

For simplicity,



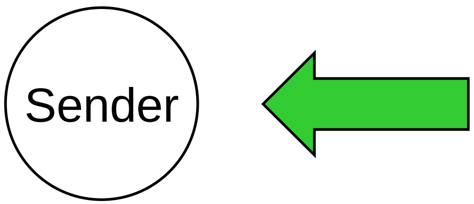
Pseudo-dimension $k=t$

Pseudo-basis $B=\{Y_1, \ldots, Y_t\}$

S computes

$\{E_i=Y_i-X_i \mid Y_i \in B\}$

For simplicity,



Pseudo-dimension $k=t$

Pseudo-basis $B=\{Y_1, \ldots, Y_t\}$

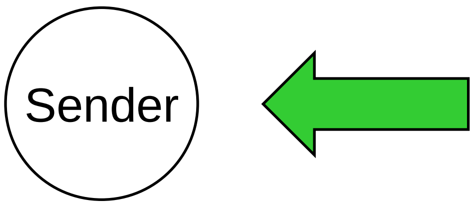
S computes

$$\{E_i = Y_i - X_i \mid Y_i \in B\}$$

$$= \text{basis of } [E_1, \ldots, E_{t+1}]$$

from the definition of pesudo-basis

For simplicity,



Pseudo-dimension $k=t$

Pseudo-basis $B=\{Y_1, \ldots, Y_t\}$

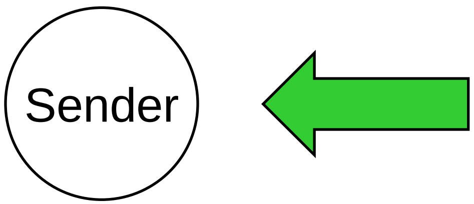
S computes

$$\{E_i = Y_i - X_i \mid Y_i \in B\}$$

$$= \text{basis of } [E_1, \ldots, E_{t+1}]$$

$$\text{FORGED} = \cup \text{ NonZero(these } E_i \text{)}$$

For simplicity,



Pseudo-dimension $k=t$

Pseudo-basis $B=\{Y_1, \ldots, Y_t\}$

S computes

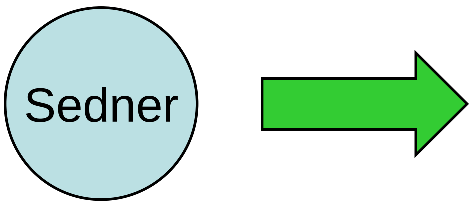
$\{E_i=Y_i-X_i \mid Y_i \in B\}$

$= \text{basis of } [E_1, \ldots, E_{t+1}]$

FORGED $= \cup$ NonZero(these E_i)

$= \{$ all forged channels $\}$

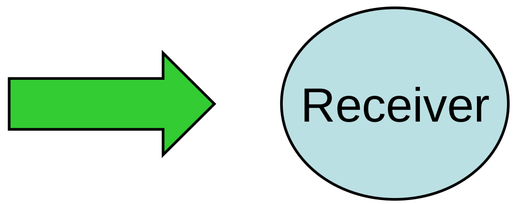
In the 3rd round



FORGED

$c = s + f_{t+1}(0)$

where $Y_{t+1} \notin B$



R decrypts c as follows.

R received FORGED

Suppose that FORGED={1, …, t}

$$X_{t+1}=(\overbrace{f_{t+1}(1), \ldots, f_{t+1}(t)}^{\text{R ignores}}, \overbrace{f_{t+1}(t+1), \ldots, f_{t+1}(n)}^{\text{R received these } t+1 \text{ values correctly}})$$

Perfect Reliability

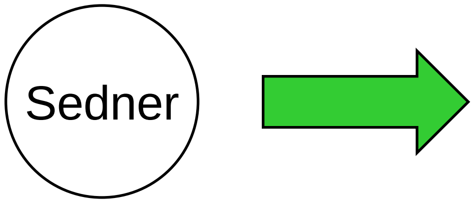
$$X_{t+1}=(f_{t+1}(1), \ldots, f_{t+1}(t), \underbrace{f_{t+1}(t+1), \ldots, f_{t+1}(n)})$$

R can reconstruct $f_{t+1}(x)$ from these $t+1$ by using Lagrange formula.

Therefore R can decrypt

$$c = s + f_{t+1}(0)$$

Perfect Privacy



FORGED

$c = s + f_{t+1}(0)$

$$X_{t+1}=(\underbrace{f_{t+1}(1), \ldots, f_{t+1}(t)}, \; f_{t+1}(t+1), \ldots, f_{t+1}(n))$$

Enemy knows at most t values.

Hence

it has no info. on $f_{t+1}(0)$.

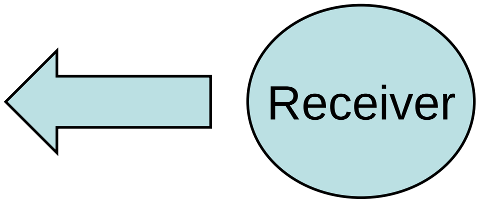
Therefore it has no info. on s.

Rest of This Talk

- Our 3-round PSMT
- Basic 2-round PSMT
- More Efficient 2-round PSMT
- Final 2-round PSMT

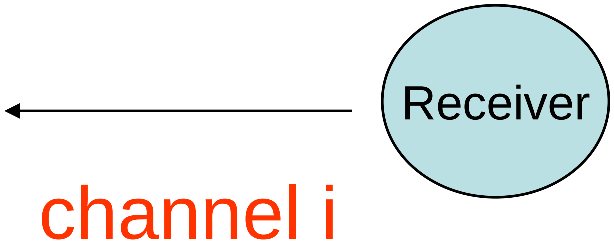
For i=1, …, n

$X_i=(f_i(1), \ldots, f_i(n))$



the coefficients of $f_i(x)$

$X_i=(f_i(1), \ldots, f_i(n))$



For i=1, …, n

Sender

$$Y_i = X_i + E_i$$

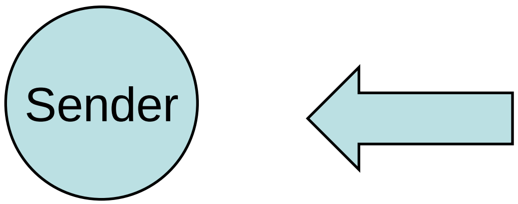
Sender

channel i

$$f_i'(x)$$

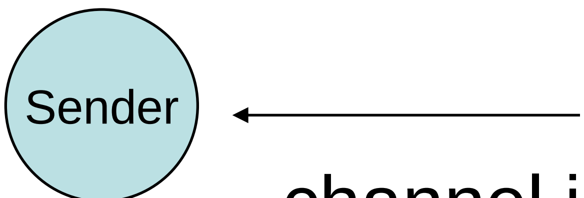
$$X_i' = (f_i'(1), \ldots, f_i'(n))$$

For i=1, …, n



$Y_i = X_i + E_i$

Note that $d(Y_i, X_i) \leqq t$



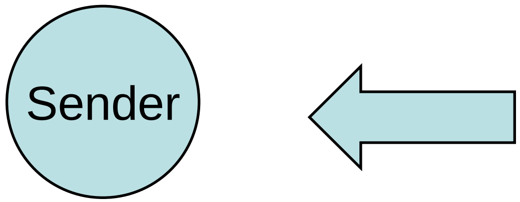
channel i

$f_i'(x)$

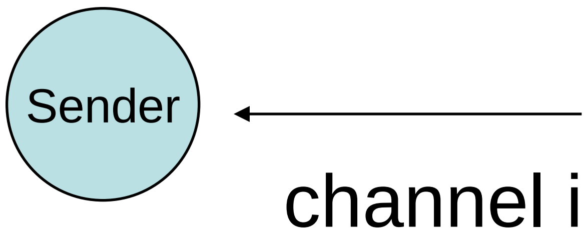
$X_i' = (f_i'(1), \ldots, f_i'(n))$

If $d(Y_i, X_i') > t$,

then S broadcasts "ignore channel i"



$Y_i = X_i + E_i$



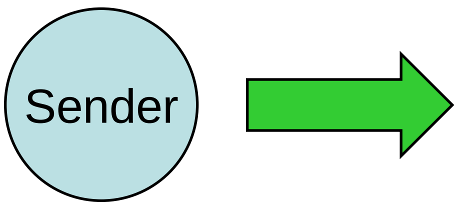
$f_i'(x)$

$X_i' = (f_i'(1), \ldots, f_i'(n))$

If $d(Y_i, X_i') > t$,
then S broadcasts "ignore channel i"

Otherwise
S broadcasts $\Delta_i = X_i' - Y_i$

In the 2nd round



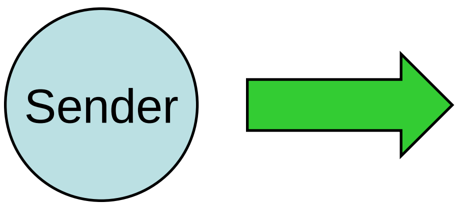
Each Δ_i

Pseudo-dimension k

Pseudo-basis B

$c=s+f_1'(1)+\ldots+f_n'(n)$

In the 2nd round



Each Δ_i

Pseudo-dimension k

Pseudo-basis B

$c = s + f_1'(1) + \ldots + f_n'(n)$

R first computes FORGED.

R next reconstrcuts each $f_i'(x)$ as follows.

For each $j \notin$ FORGED,

- R computes

$$f_i'(j) = \Delta_i|_j + f_i(j)$$

$$= (X_i' - Y_i)|_j + f_i(j)$$

- This holds because

$$f_i'(j) = X_i'|_j \text{ and } Y_i|_j = f_i(j)$$

For each $j \notin$ FORGED,

- R computes

$$f_i'(j) = \Delta_i \, l_j + f_i(j)$$

$$= (X_i' - Y_i)\, l_j + f_i(j)$$

- This holds because

$$f_i'(j) = X_i' l_j \text{ and } y_{ij} = f_i(j)$$

- R can reconstrcut $f_i'(x)$ from these $f_i'(j)$ by using Lagrange formula.

Perfect Reliability

Thus R can reconstrcut each $f_i'(x)$.

Hence R can decrypt

$$c = s + f_1'(1)+\ldots+f_n'(n)$$

Perfect Privacy

- S broadcasts a pseudo-basis $\{Y_1, \ldots, Y_t\}$
- Enemy corrupts t channels.
- Note that

$$n - t - t = (2t+1) - t - t = 1$$

- This implies that
  there remains at least one $f_i'(i)$
  on which the enemy has no information

Perfect Privacy

- Hence in the ciphertext

  $c = s + f_1'(1) + \ldots + f_n'(n)$,

- the enemy has no information on s.
- Hence

  perfect privacy is also satisfied.

Efficiency

	Trans. rate	Sender's Comp.	Receiver's Comp.
Basic scheme	$O(n^2 t)$	poly(n)	poly(n)
More efficient scheme	$O(n^2)$	poly(n)	poly(n)
Final scheme	$O(n)$	poly(n)	poly(n)

Efficiency

	Trans. rate	Sender's Comp.	Receiver's Comp.
Basic scheme	$O(n^2t)$	poly(n)	poly(n)
More efficient scheme	$O(n^2)$	poly(n)	poly(n)
Final scheme	$O(n)$	poly(n)	poly(n)

More Efficient 2-round PSMT

- In our basic scheme,

  S sends a single secret s.

More Efficient 2-round PSMT

- In our basic scheme,
  S sends a single secret s.
- In the more efficient scheme,
  S sends t^2 secrets s_i by running
  the basic scheme t times in parallel.

More Efficient 2-round PSMT

- In our basic scheme,
  S sends a single secret s.
- In the more efficient scheme,
  S sends t^2 secrets s_i by running
  the basic scheme t times in parallel.

This implies that the transmission rate is reduced from $O(n^2t)$ to $O(n^2)$.

Run the basic scheme t times

- For each channel i,
  R chooses t polynomials $f_{i+jn}(x)$,
  where $j=0, \ldots, t-1$.
- In total,
  R chooses tn polynomials $f_{i+jn}(x)$.

Among tn polynomials $f_{i+jn}(x)$,

- Since the enemy corrupts t channels, she knows t^2 values of $f_{i+jn}(i)$.

Among tn polynomials $f_{i+jn}(x)$,

- Since the enemy corrupts t channels, she knows t^2 values of $f_{i+jn}(i)$.
- S broadcasts a pseudo-basis $\{Y_1, \ldots, Y_t\}$

Among tn polynomials $f_{i+jn}(x)$,

- Since the enemy corrupts t channels, she knows t^2 values of $f_{i+jn}(i)$.
- S broadcasts a pseudo-basis $\{Y_1, \ldots, Y_t\}$
- There remains t^2 uncorrupted f_{i+jn}'(i)s because

$$tn - t^2 - t = t(2t+1) - t^2 - t = t^2$$

Enemy has no info. on these t^2 values

Randomness Extractor

- is used to extracst these t^2 values
- S uses them as one-time pad
  to encrypt t^2 secrets

Randomness Extractor

- Suppose that Enemy has no info. on t^2 out of tn elements $r_0, \ldots, r_{tn-1}$.
- Let

  $$R(x)=r_0+r_1x+\ldots + r_{tn-1}\, x^{tn-1}$$

- Then Enemy has no info. on

  $$R(1), \ldots, R(t^2)$$

Consequently,

- In the more efficient scheme,
  S can send t^2 secrets s_i by running
  the basic scheme t times in parallel.

  This implies that the transmission rate is
  reduced from $O(n^2t)$ to $O(n^2)$.

Efficiency

	Trans. rate	Sender's Comp.	Receiver's Comp.
Basic scheme	$O(n^2t)$	poly(n)	poly(n)
More efficient scheme	$O(n^2)$	poly(n)	poly(n)
Final scheme	$O(n)$	poly(n)	poly(n)

Efficiency

	Trans. rate	Sender's Comp.	Receiver's Comp.
Basic scheme	$O(n^2t)$	poly(n)	poly(n)
More efficient scheme	$O(n^2)$	poly(n)	poly(n)
Final scheme	$O(n)$	poly(n)	poly(n)

Most Costly Part

- S broadcasts $\Delta_1, \ldots \Delta_{tn}$, where $|\Delta_i| \leqq t$.
- The communication cost
  to broadcast each Δ_i is tn.
- We will show how to reduce it to O(n).

Modify the 2nd round as follows.

- S first computes the pseudo-dimension k.
- If $|\Delta_i|>k$,

  S broadcasts "ignore channel i".

Otherwise S sends Δ_i as follows

- $|\Delta_i| \leqq k$
- S knows the pseudo-dimension k.
- R knows FORGED={k forged channels}

Generalized Broadcast

- Suppose that S wants to send k+1 elements $a_0, \ldots, a_k$.
- S constructs A(x) such that

  $$A(x)= a_0 + a_1x+ \ldots + a_kx^k$$

- S sends A(i) throught channel i for i=1, …,n.
- This communication cost is n.

R receives as follows.

- Suppose that FORGED=$\{1, \ldots, k\}$.
- R ignores FORGED and considers a shortened codeword

$$[A(k+1), \ldots, A(n)]$$

- It turns out that

$$d = 2(t - k) + 1$$

R receives as follows.

- Hence R can correct t-k errors.
- On the other nhand,

  since there are k forged channels,

  Enemy can forge more t-k channels.
- Therefore

  R can receive $a_0, \ldots, a_k$ correctly.

Transmission Rate

- By using this technique,
  the cost of sending each Δ_i is
  reduced from tn to n.
- This implies that the transmission rate is reduced from $O(n^2)$ to $O(n)$.

Efficiency

	Trans. rate	Sender's Comp.	Receiver's Comp.
Basic scheme	$O(n^2t)$	poly(n)	poly(n)
More efficient scheme	$O(n^2)$	poly(n)	poly(n)
Final scheme	$O(n)$	poly(n)	poly(n)

Summary

- We solved the open problem raised by Agarwal, Cramer and de Haan at CRYPTO 2006.

2-round PSMT for n=2t+1

	Trans. rate	Sender's comp.	Receiver's comp.
Agarwal et al.'s schme	O(n)	exponential	exponential
Proposed scheme	O(n)	poly(n)	poly(n)

Thank you !