TLS1.2

簡単なハンドシェイク演習

大津 繁樹

XXXX年X月X日

演習道具の確認

演習用ポストイット３色

筆記用具

TLSハンドシェイク(full handshake)

クライアント

ClientHello

ServerHello
Certificate
ServerKeyExchange
ServerHelloDone

サーバ

ClientKeyExchange
ChangeCipherSpec
Finished

ChangeCipherSpec
Finished

Application Data

TLSハンドシェイクの意味

ClientHello/ServerHello/ServerHelloDone
TLSのための情報交換

バージョン・乱数・暗号方式・拡張情報

Certificate
公開鍵情報の送付

エンドポイントの認証

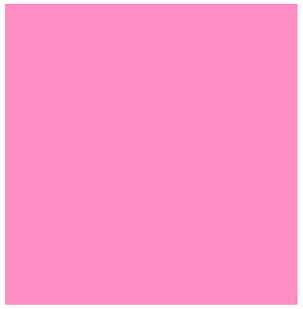
ClientKeyExchange/ServerKeyExchange
共有鍵交換

ChangeCipherSpec
暗号開始の合図

Finished
ハンドシェイクデータの改ざんチェック

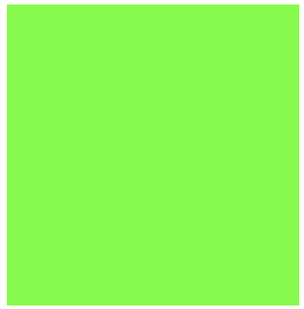
演習用ポストイット

クライアントが送信する
パケット



サーバが送信する
パケット

暗号化された
パケット
(サーバ・クライアント共通)

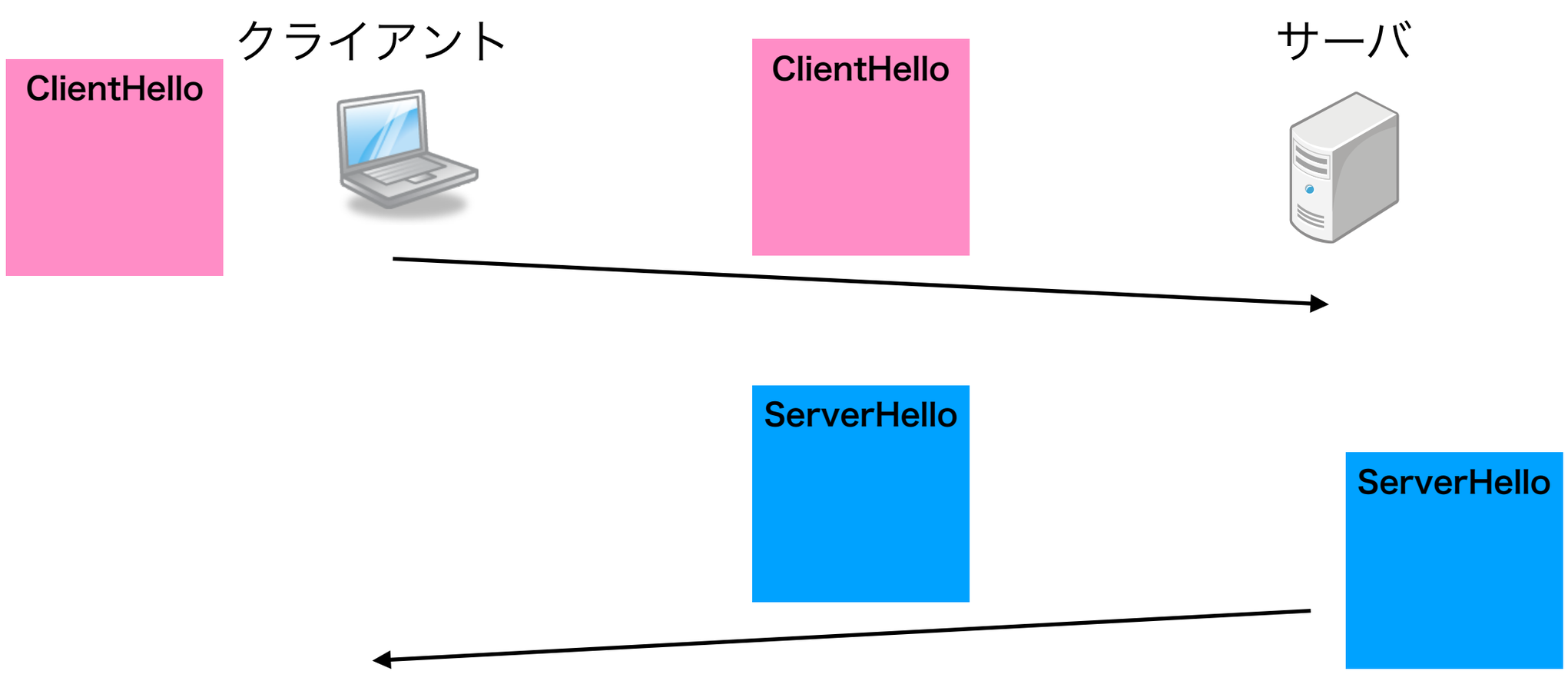


２名１組になってクライアント役、サーバ役を決めてください。

ポストイットによる
ハンドシェイクパケット

ClientHello ← ハンドシェイクのタイプ名

まずは中身が空でTLSハンドシェイクのやり取り



クライアントは、２枚の赤のポストイットにClientHelloを書いて１枚をサーバに渡してください（中のデータ部分は空で構いません。）

サーバ役の人は、クライアントからClientHelloをもらったら、同様に青色のServerHelloを２枚書いて、１枚をクライアントに渡してください。

お題１に従ってクライアントが
ChangeChipherSpecを送るまで続けて送る
(必ず２枚書いて１枚を手元に)

ClientHello

ServerHello

Certificate

ServerKeyExchange

ServerHelloDone

ClientHello

ServerHello

ClientKeyExchange

ChangeCipherSpec

ここで一旦ストップ

ClientHello

ClientHello

ServerHello

ServerHello

Certificate

Certificate

ServerKeyExchange

ServerKeyExchange

ServerHelloDone

ServerHelloDone

ClientKeyExchange

ClientKeyExchange

ChangeCipherSpec

ChangeCipherSpec

Finishedの作り方

Finished

これまでやり取りした文字数(記号やスペースを除く、数字は１文字分)

ClientHello(11文字)のようにこれまでやり取りしたハンドシェイクの文字数を合計して、その数を Finished に書いて送ります。 送付するFinished自身は計算対象に含まれません。

(* 実際はChangeCipherSpecの文字数は抜かしますが、今回は簡単のため入れて計算してください。)

文字数早見表

ハンドシェイク名	文字数
ClientHello	11
ServerHello	11
Certificate	11
ServerKeyEchange	17
ServerHelloDone	15
ClientKeyExchange	17
ChangeCipherSpec	16
Finished	8

ポストイットの裏に各ハンドシェイクパケットの文字数を書いておくといいでしょう。

クライアントからFinishedを送る

11
+
11
+
11
+
17
+
15
+
17
+
16
=
98

ここまでの文字数を計算

ClientHello

ServerHello

Certificate

ServerKeyExchange

ServerHelloDone

ClientKeyExchange

ChangeCipherSpec

Finished
文字数

Finished

これまでやり取りした文字数(記号やスペースを除く、数字は１文字分)

ClientHello

ServerHello

Certificate

ServerKeyExchange

ServerHelloDone

ClientKeyExchange

ChangeCipherSpec

サーバは、Finishedの中身をチェック

ClientHello

ServerHello

Certificate

ServerKeyExchange

ServerHelloDone

ClientKeyExchange

ChangeCipherSpec

Finished
98

ClientHello 11

\+

ServerHello 11

\+

Certificate 11

\+

ServerKeyExchange 17

\+

ServerHelloDone 15

\+

ClientKeyExchange 17

\+

ChangeCipherSpec 16

=

98

ここまでの文字数

一致している？

改ざんされていないことをチェック

Finished
98

サーバから残りを送る

ClientHello

ServerHello

Certificate

ServerKeyExchange

ServerHelloDone

ClientKeyExchange

ChangeCipherSpec

Finished

Finished

これまでやり取りした文字数(記号やスペースを除く、数字は１文字分)

ChangeCipherSpec

Finished

ここまでの文字数を計算

ClientHello

ServerHello

Certificate

ServerKeyExchange

ServerHelloDone

ClientKeyExchange

ChangeCipherSpec

Finished

ChangeCipherSpec

11 + 11 + 11 + 17 + 15 + 18 + 17 + 9 (数字は１文字分) + 16 = 123

クライアントは、Finishedの中身をチェック

ClientHello 11
+
ServerHello 11
+
Certificate 11
+
ServerKeyExchange 17
+
ServerHelloDone 15
+
ClientKeyExchange 16
+
ChangeCipherSpec 16
+
Finished 10 (数字は１文字分)
+
ChangeCipherSpec 17
=
Finished

ここまでの文字数を計算して、受け取ったFinishedの数と合っているかチェック

Finishedのチェックが終われば
TLSハンドシェイク完了

ClientHello
ServerHello
Certificate
ServerKeyExchange
ServerHelloDone
ClientKeyExchange
ChangeCipherSpec
Finished
ChangeCipherSpec
Finished

サーバ・クライアント入れ替え

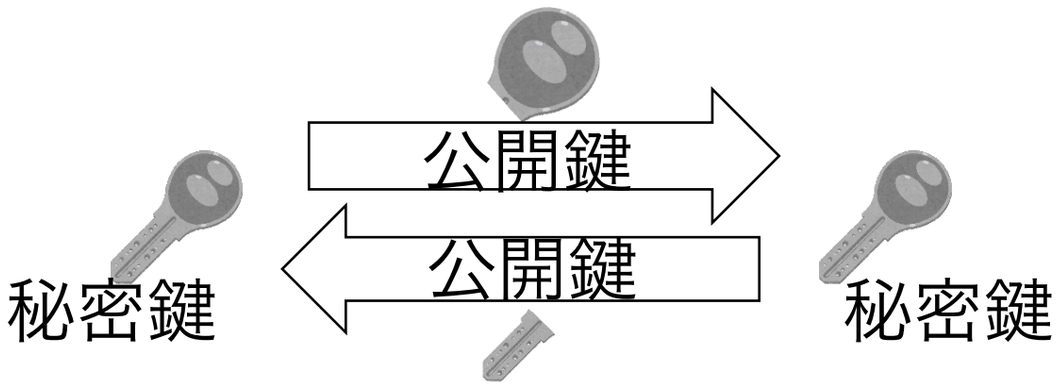
- サーバ役、クライアント役の人を入れ替えて再度演習しましょう。

一旦置いておく

これだけだと不十分

この後の演習でまた利用します。

鍵交換



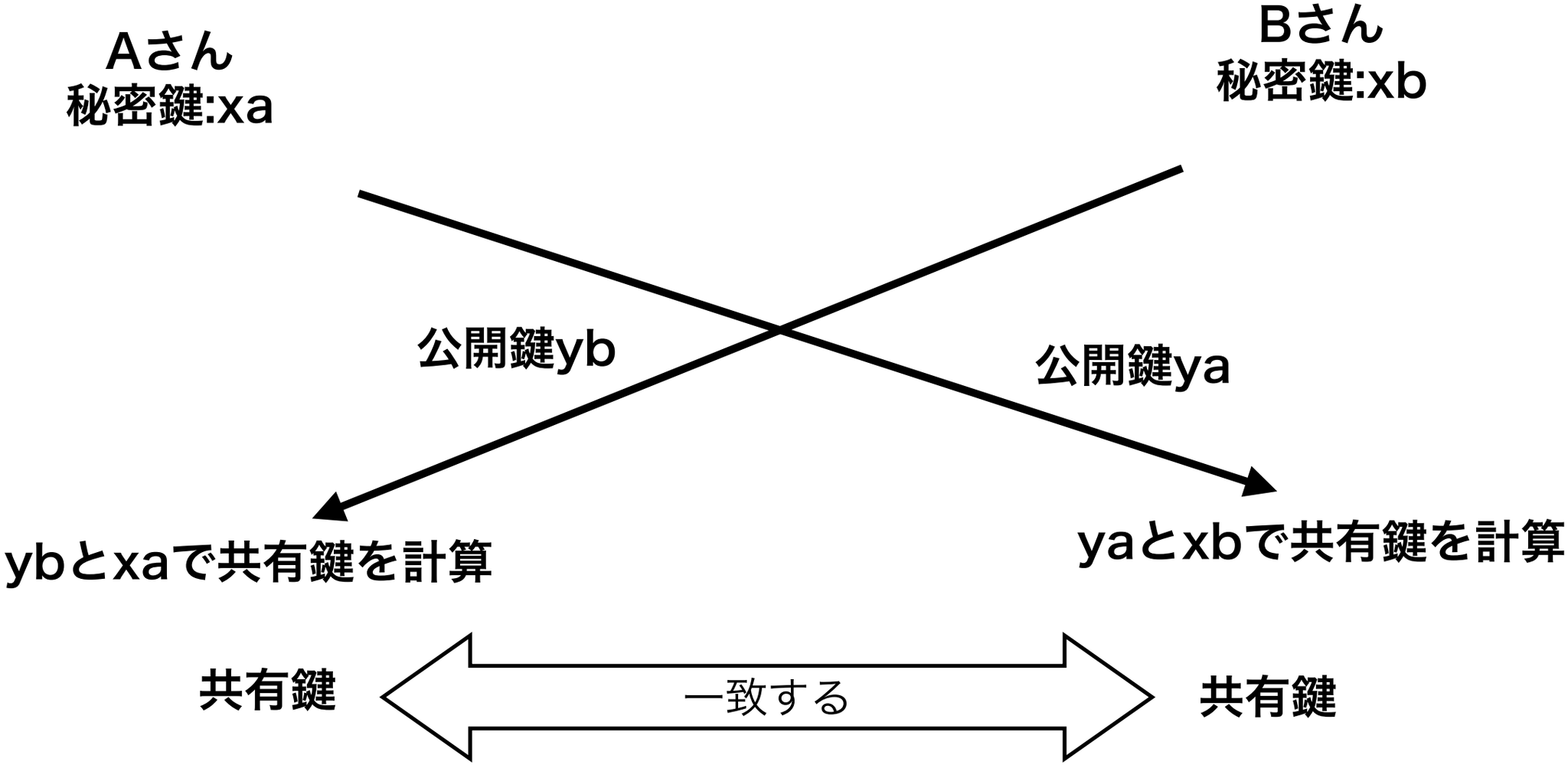
- 2者間で安全に鍵を共有する仕組み
- 互いに公開鍵を交換しあい、共有鍵を生成する。
- 通信経路上で共有鍵のやり取りがない。

- DH (Diffie-Hellman)
- ECDH(楕円曲線DH)

鍵交換演習

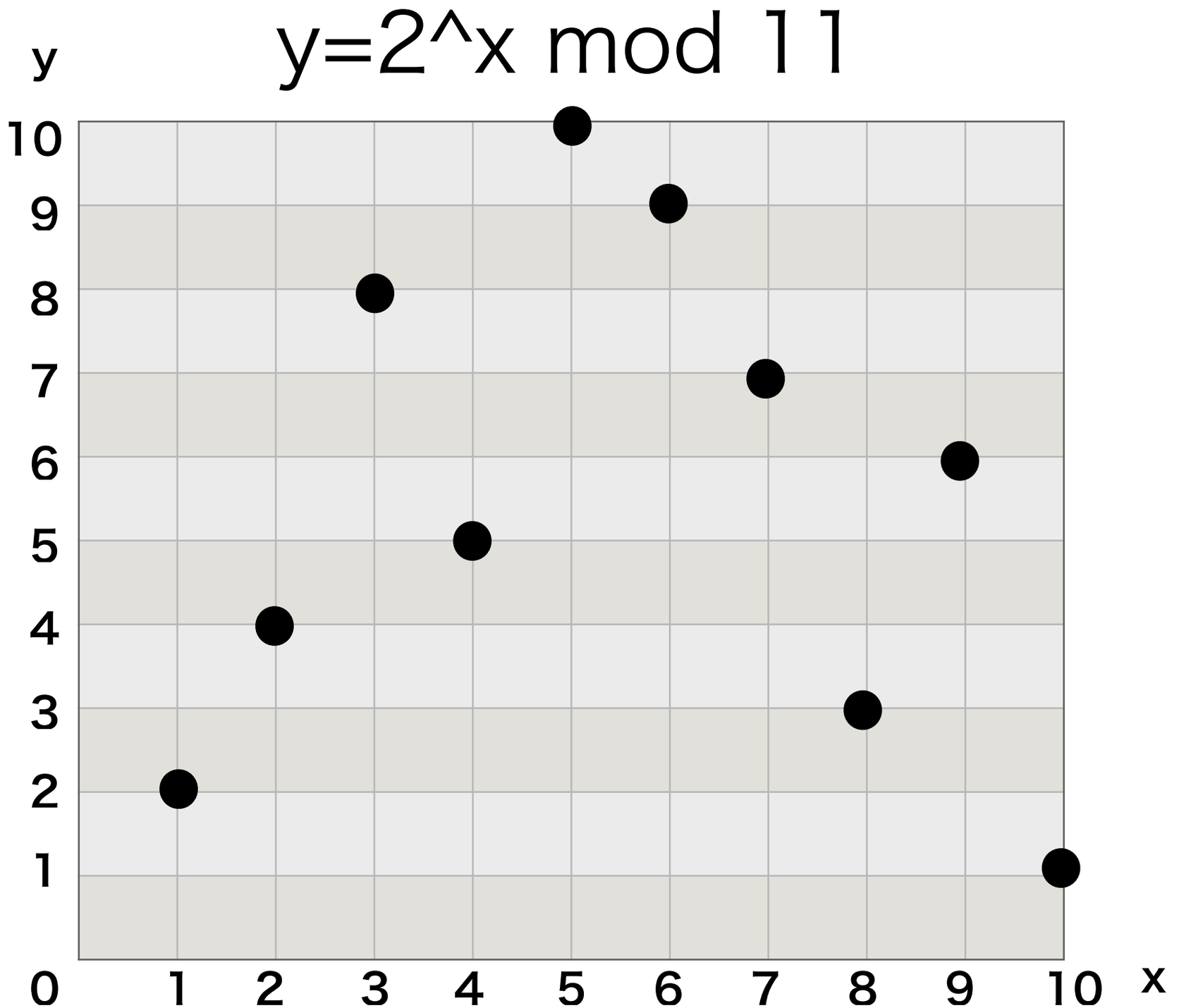
後のTLSハンドシェイク演習でも使います。

Diffe-Hellman鍵交換



相手からもらった公開鍵と自分の秘密鍵を組み合わせて A、B共有の鍵を作る

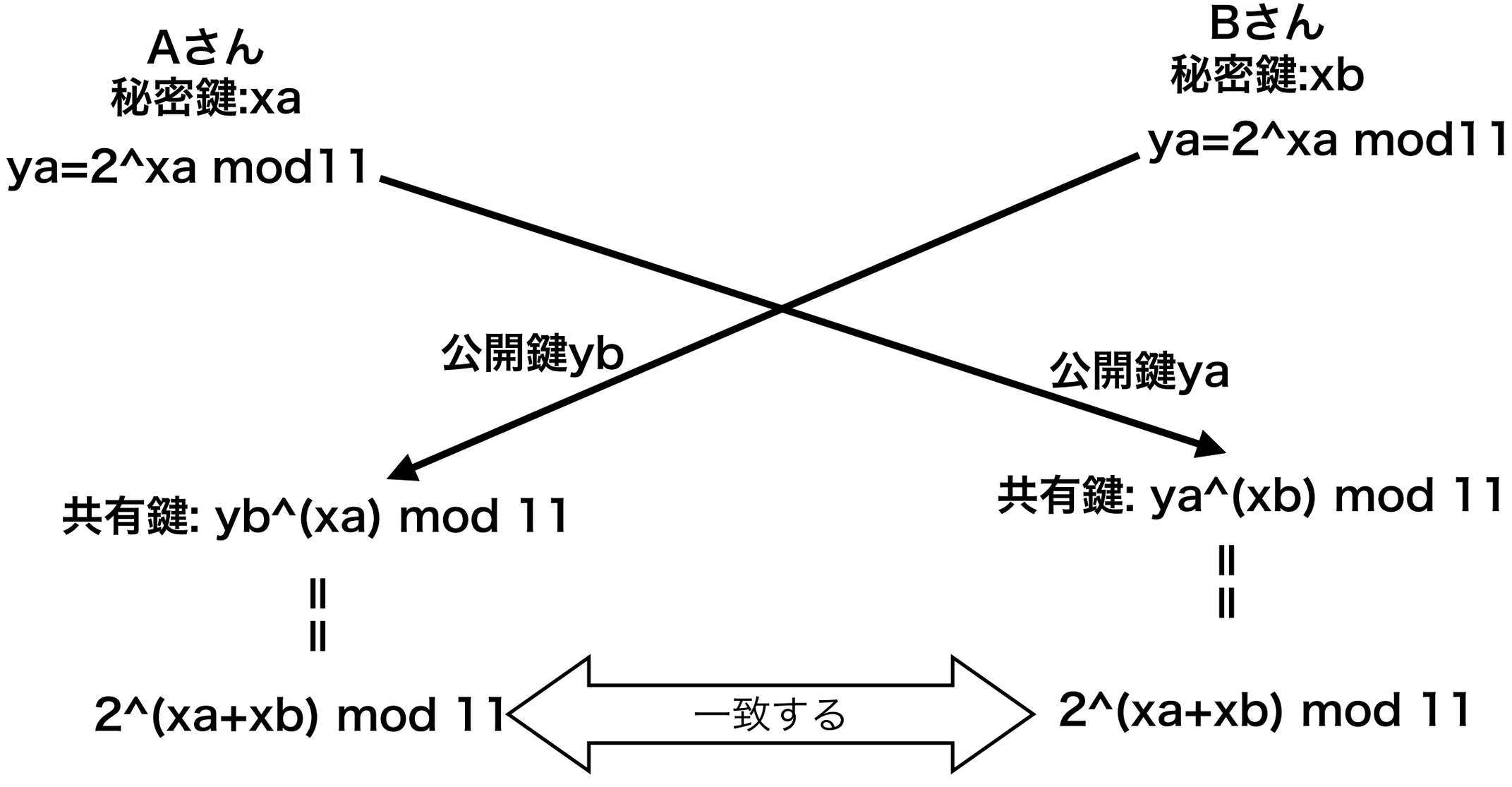
離散対数問題



xからyを計算することはできるが、yからxを計算するのは困難

x: 秘密鍵, y: 公開鍵

離散対数問題を用いた
Diffe-Hellman鍵交換



演習

- 二人一組になる。
- 1から10の間で好きな数字xaを選ぶ(秘密鍵)
- 計算表使って ya=2^xa mod 11 の公開鍵 ya を求める。
- 自分の公開鍵yaを相手に渡す。相手の公開鍵ybをもらう。
- 計算表を使って (yb)^xa mod 11 の共有鍵を求める。
- せーので共有鍵を言い合う。同じ数字ならOK。

離散対数式の計算表

y = g^x mod 11

公開鍵を求める列

g \ x	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	8	5	10	9	7	3	6	1
3	3	9	5	4	1	3	9	5	3	1
4	4	5	9	3	1	4	5	9	3	1
5	5	3	4	9	1	5	3	4	9	1
6	6	3	7	9	10	5	8	4	2	1
7	7	5	2	3	10	4	6	9	8	1
8	8	9	6	4	10	3	2	5	7	1
9	9	4	3	5	1	9	4	3	5	1
10	1	10	1	10	1	10	1	10	1	10

演習解答例(1)

- 二人一組になる。
- 1から10の間で好きな数字xaを選ぶ(秘密鍵)

一例：　**秘密鍵xa: 3**

演習解答例(2)

- 計算表使って ya=2^xa mod 11 の公開鍵 ya を求める。

g^x	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	8	5	10	9	7	3	6	1
3	3	9	5	4	1	3	9	5	3	1
4	4	5	9	3	1	4	5	9	3	1
5	5	3	4	9	1	5	3	4	9	1
6	6	3	7	9	10	5	8	4	2	1
7	7	5	2	3	10	4	6	9	8	1
8	8	9	6	4	10	3	2	5	7	1
9	9	4	3	5	1	9	4	3	5	1
10	1	10	1	10	1	10	1	10	1	10

公開鍵 ya: 8

演習解答例(3)

- 自分の公開鍵yaを相手に渡す。相手の公開鍵ybをもらう。

- 計算表を使って (yb)^xa mod 11 の共有鍵を求める。

g^x	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	2	4	8	5	10	9	7	3	6	1
3	3	9	5	4	1	3	9	5	3	1
4	4	5	9	3	1	4	5	9	3	1
5	5	3	4	9	1	5	3	4	9	1
6	6	3	7	9	10	5	8	4	2	1
7	7	5	2	3	10	4	6	9	8	1
8	8	9	6	4	10	3	2	5	7	1
9	9	4	3	5	1	9	4	3	5	1
10	10	1	10	1	10	1	10	1	10	1

相手の公開鍵 yb: 9 とする。

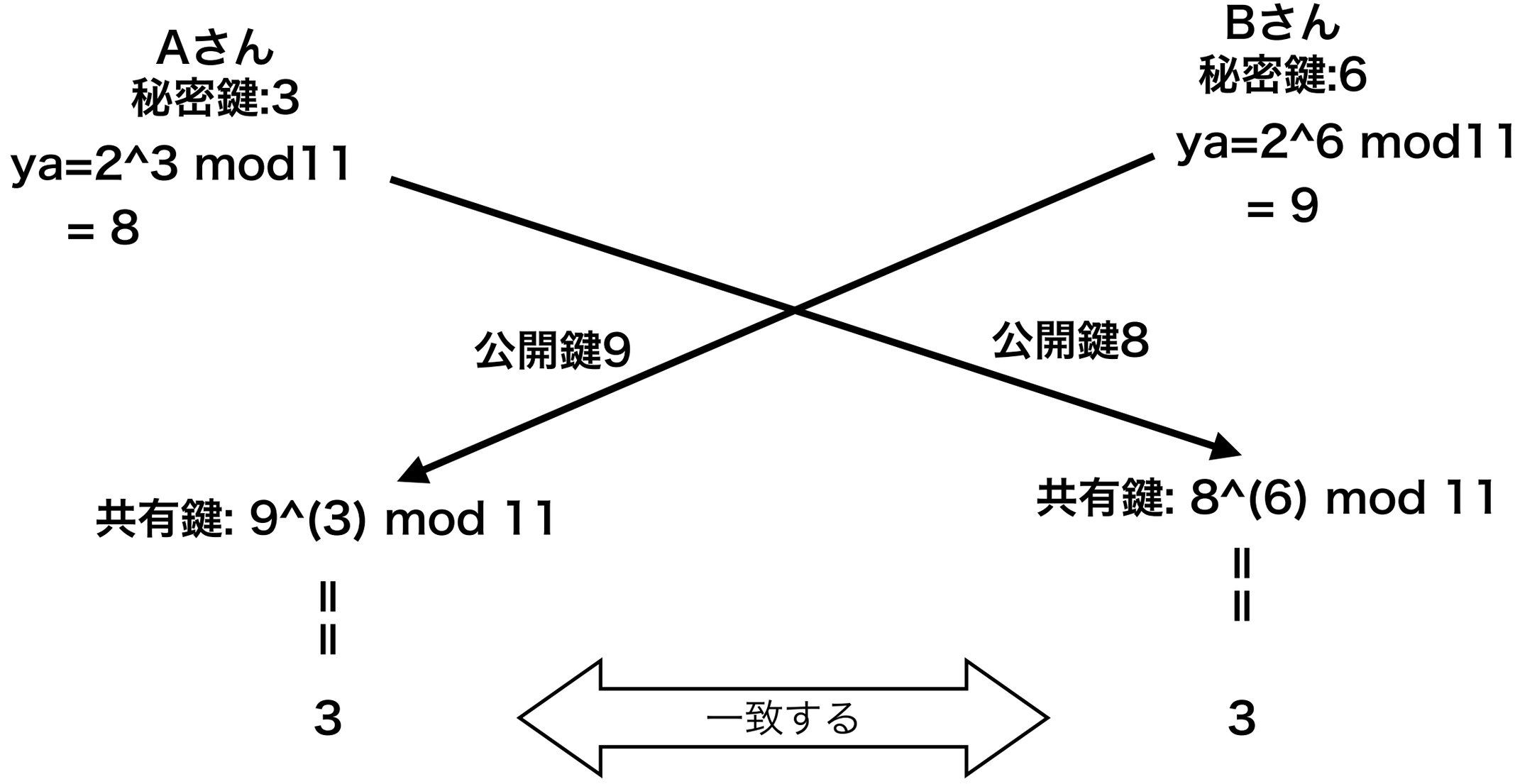
自分の秘密鍵 xa: 3 である。

共通鍵

(yb)^xa mod 11 =

9^3 mod 11 = 3 (青色)

Diffe-Hellman鍵交換演習解答



TLSの仕組み

TLSハンドシェイク(full handshake)

ClientHello

ServerHello
Certificate
ServerHelloDone

ClientKeyExchange
ChangeCipherSpec
Finished

ChangeCipherSpec
Finished

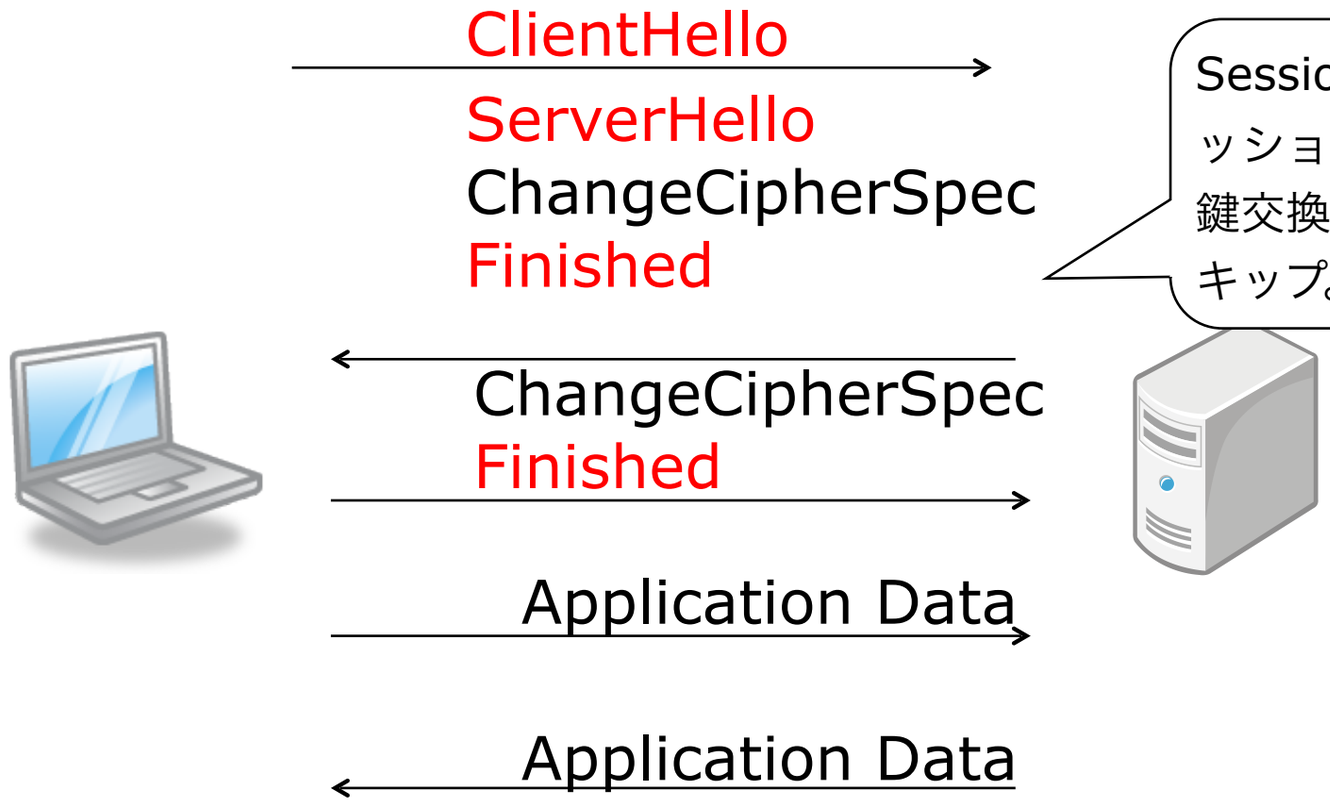
Application Data

Application Data

ClientHelloとServerHelloのやり取りで双方が利用するTLSバージョンや暗号化方式などを合意する。

(赤文字はハンドシェイク)

TLSハンドシェイク(resumption)



(赤文字はハンドシェイク)

今回は演習の対象外です

TLSハンドシェイクの意味

ClientHello/ServerHello/ServerHelloDone
TLSのための情報交換

バージョン・乱数・暗号方式・拡張情報

Certificate
サーバの認証

ClientKeyExchange/ServerKeyExchange
鍵交換

ChangeCipherSpec
暗号開始の合図

Finished
ハンドシェイクデータの改ざんチェック

実際に中身を入れた

TLSハンドシェイク演習

ClientHello/ServerHelloで暗号方式の選択

ClientHello

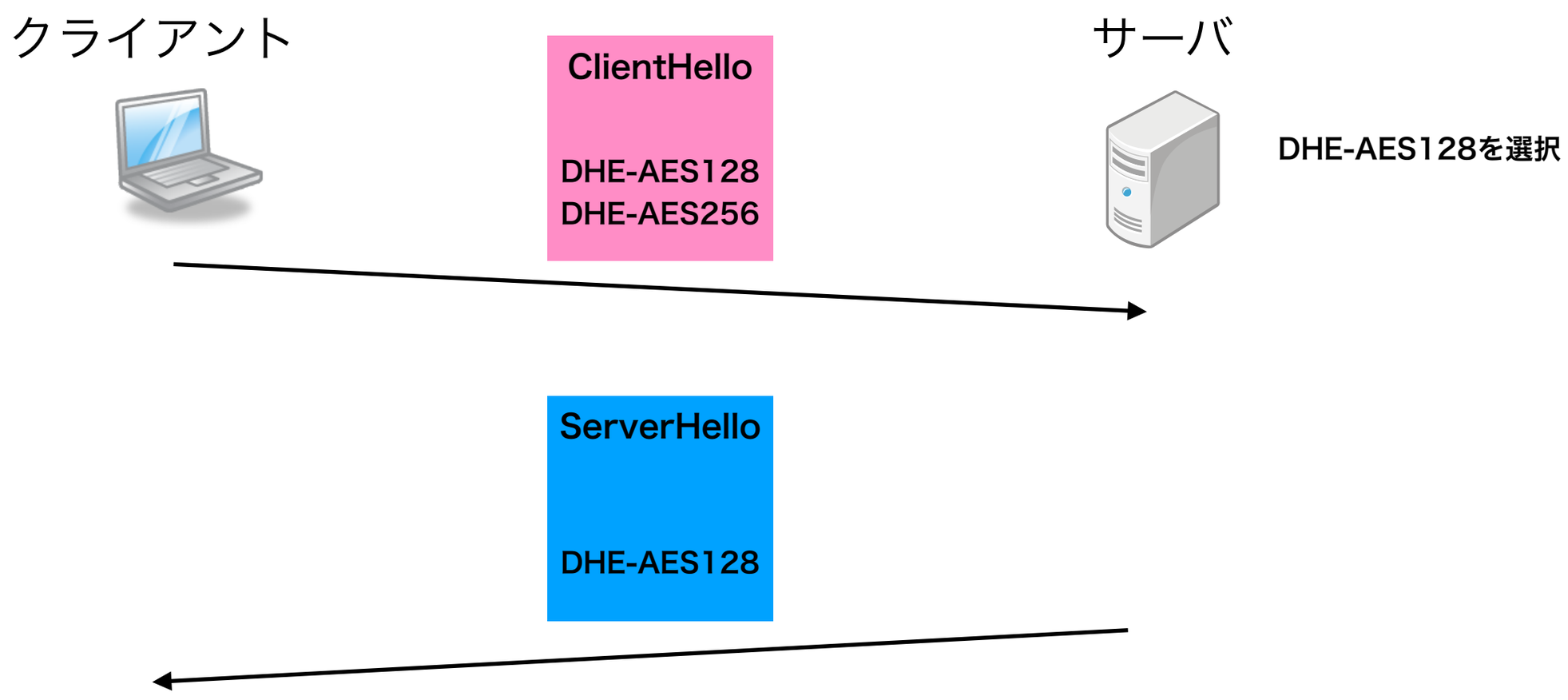
自分が使いたい複数(２つ以上)の暗号方式を書く

ServerHello

ClientHelloの中から１つ暗号方式を選択して書く

なんちゃって暗号方式リスト	記号を除いた文字数
DHE-AES128	9文字
DHE-AES256	9文字
DHE-CHACHA20	11文字

ClientHello/ServerHelloで暗号方式の選択



Certificateの中身を書いて送る

自分の名前（署名)を書きましょう。

Certificateの中身を書いて送る

クライアント

サーバ

自分の名前を書いた
Certificateを送る

Certificate

大津繁樹

通信相手の名前を
Certificateで確認

注： 本来はClientHelloに相手の名前(ホスト名)を入れます。

SeverKeyExchange
ClientKeyExchange

ServerKeyExchange

公開鍵

ClientKeyExchange

公開鍵

- 鍵交換演習と同じ方法で、秘密鍵・公開鍵を生成する
- 自分の公開鍵を書いて相手に送付
- 相手の公開鍵を受け取り、共通鍵を計算する。

鍵交換で共有鍵を求める

クライアント

公開鍵

ServerKeyExchange

8

ServerHelloDone

サーバ

秘密鍵：3

秘密鍵：6

公開鍵

ClientKeyExchange

9

共有鍵:3

共有鍵:3

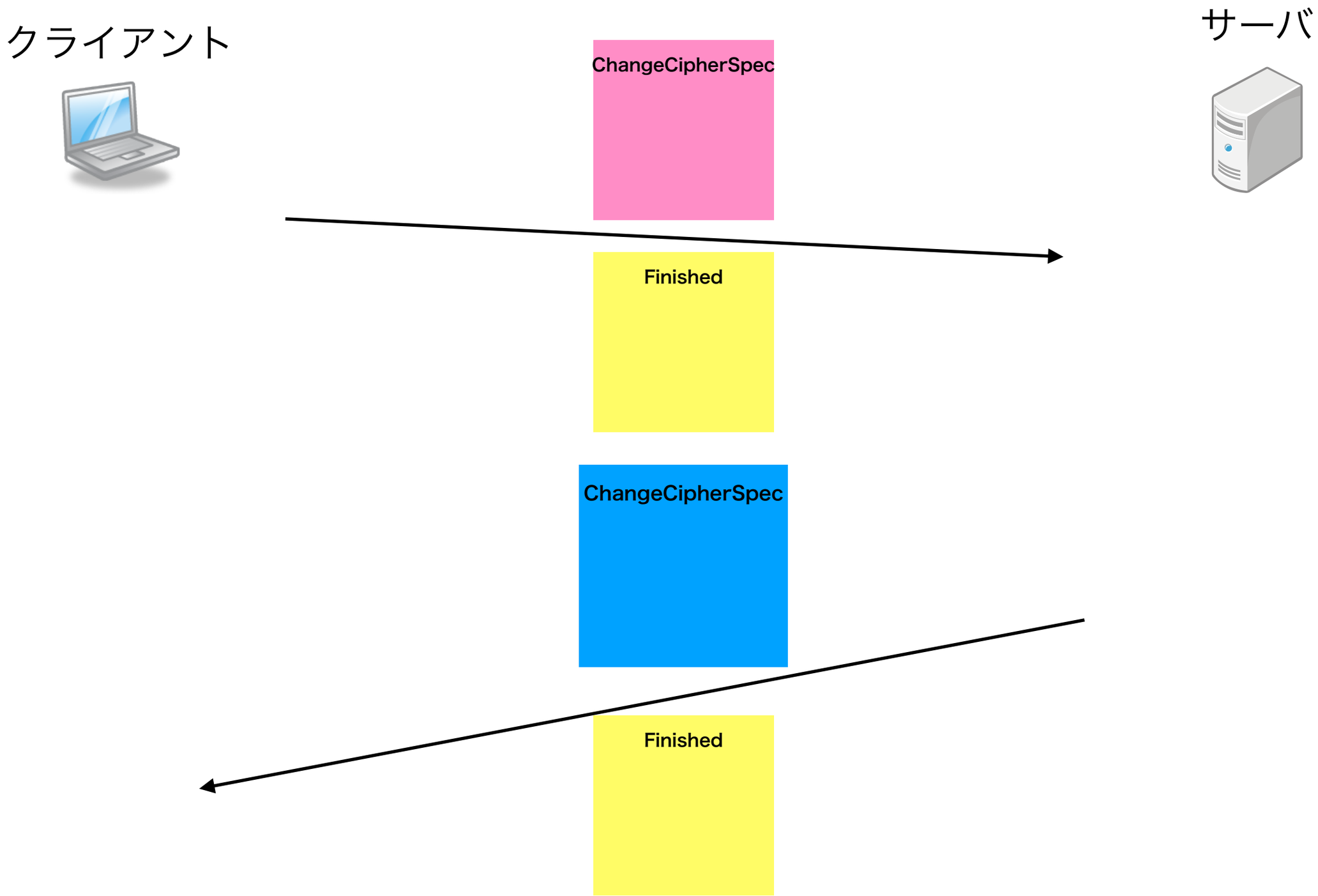
なんちゃって暗号化通信

Finished

これまでやり取りした文字数x共通鍵

- 共通鍵とFinishedのデータを掛け算して書いて送信
- 受信者は、Finishedデータを共通鍵で割り算して文字数を導出。文字数をチェックする。

なんちゃって暗号化通信



これだけでは不十分です

通信相手の顔の見えないインターネットでは、ネットワーク経路上に入ってパケットを覗き見したり改変したり、通信相手になりすましをすることが可能です。

こういった攻撃を、
Man-In-the-Middle(中間者）攻撃と呼びます。

TLSなりすまし演習1

TLSの認証の仕組みを知る

サーバなりすまし

- ３人組になってください。

クライアント役

攻撃者役

サーバ役

攻撃者は、なりすましでハンドシェイクします。

攻撃者は、サーバから送られてきたCertificateを書き写してください。
それ以外は自分でハンドシェイクを作ってください。
攻撃者は、サーバ・クライアントの２つ役割を同時に行います。

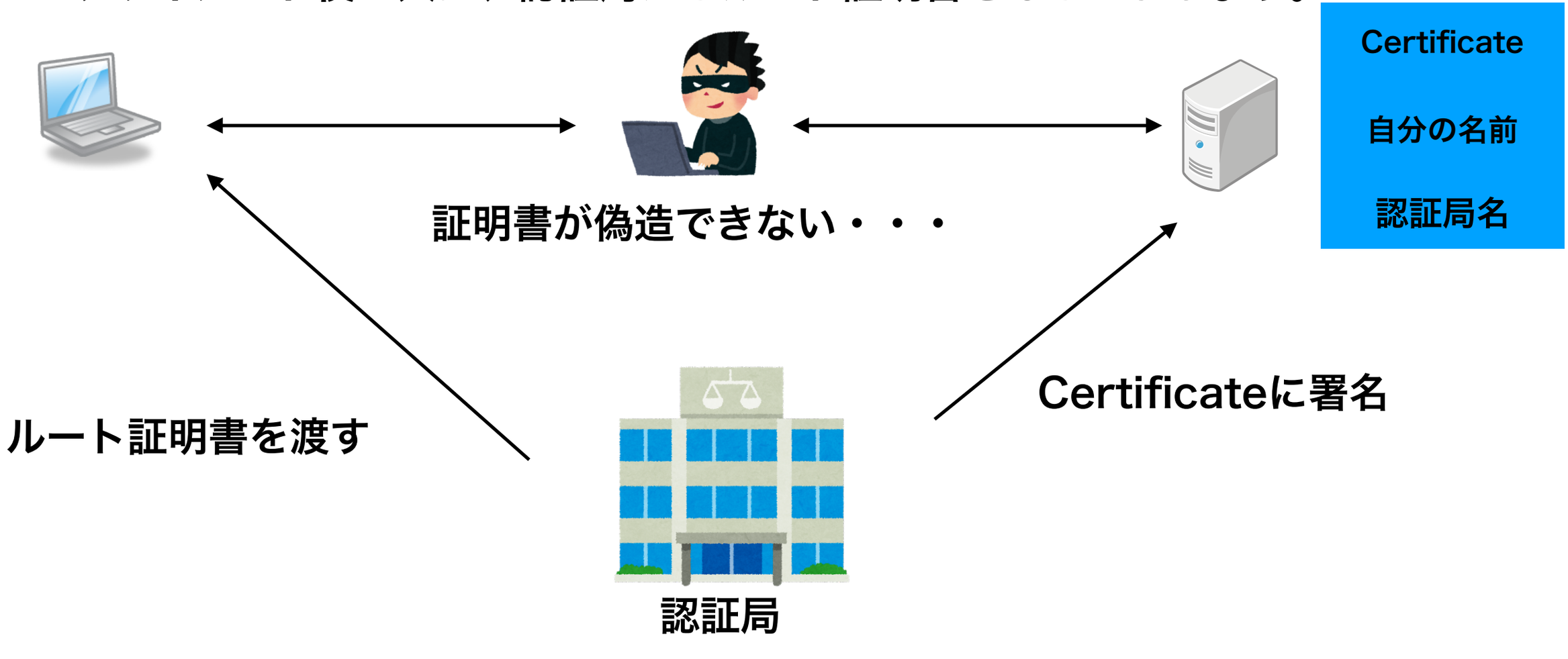
クライアントは、攻撃者がサーバになりすましているのかわかりません。

TLSなりすまし演習２

対策編

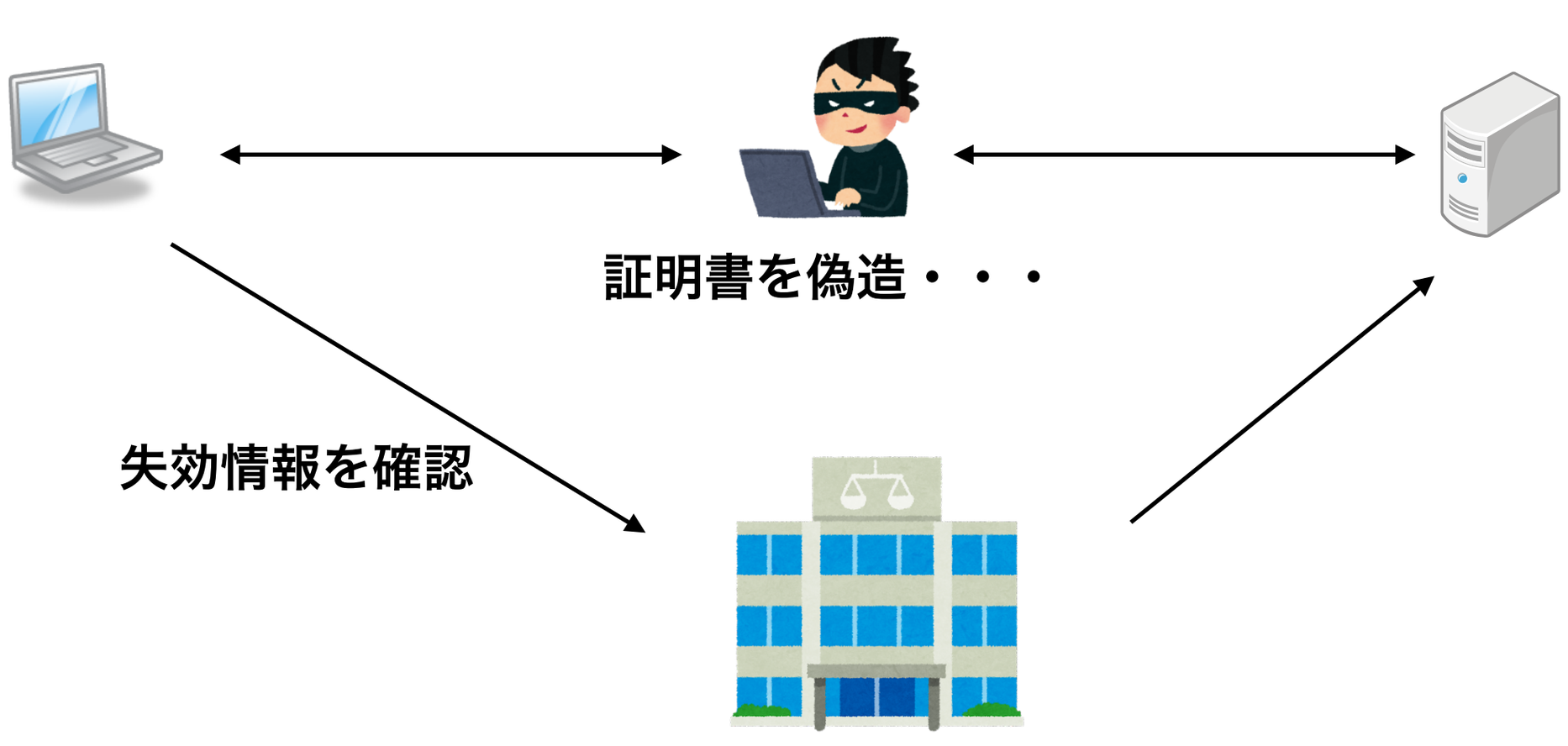
Certificateを認証局から署名してもらう

サーバ役の人は、認証局からCertificateに署名してもらいましょう。
クライアント役の人は、認証局からルート証明書をもらいましょう。



なりすましチェック

ルート証明書の署名を見比べて偽造をチェック

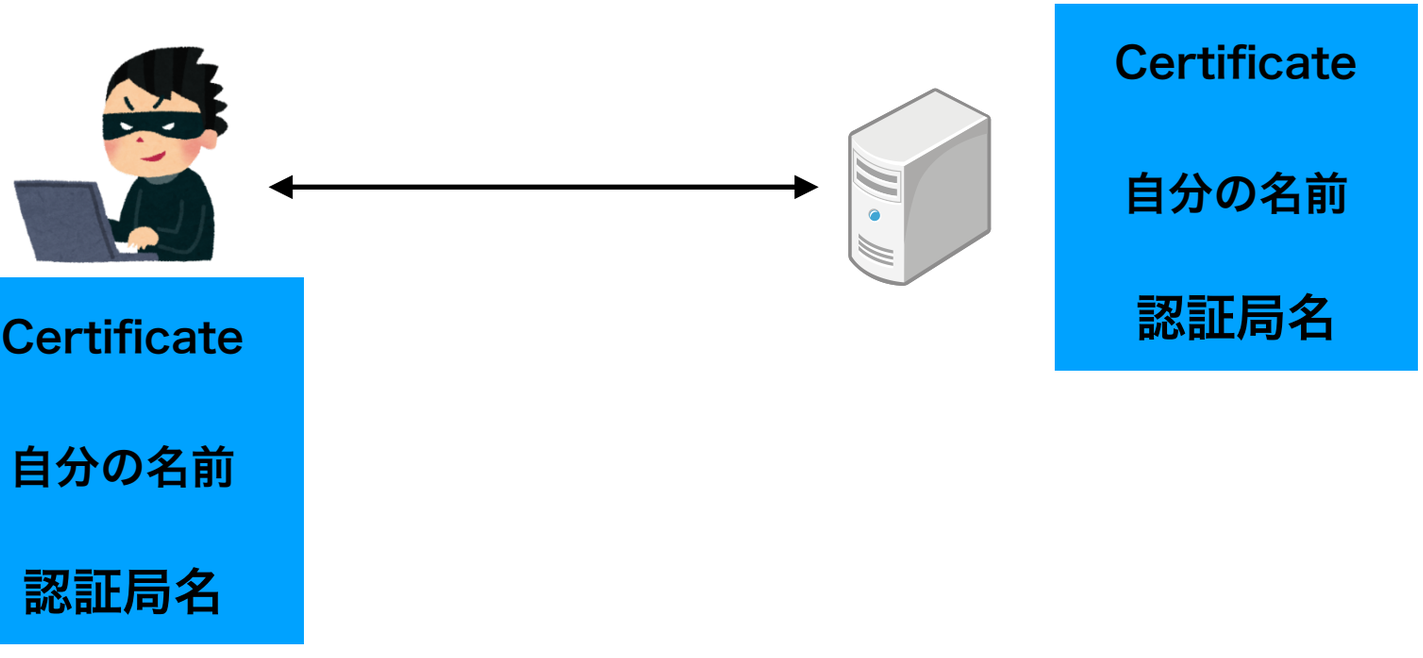


TLSなりすまし演習３

(証明書を送るだけでは不十分)

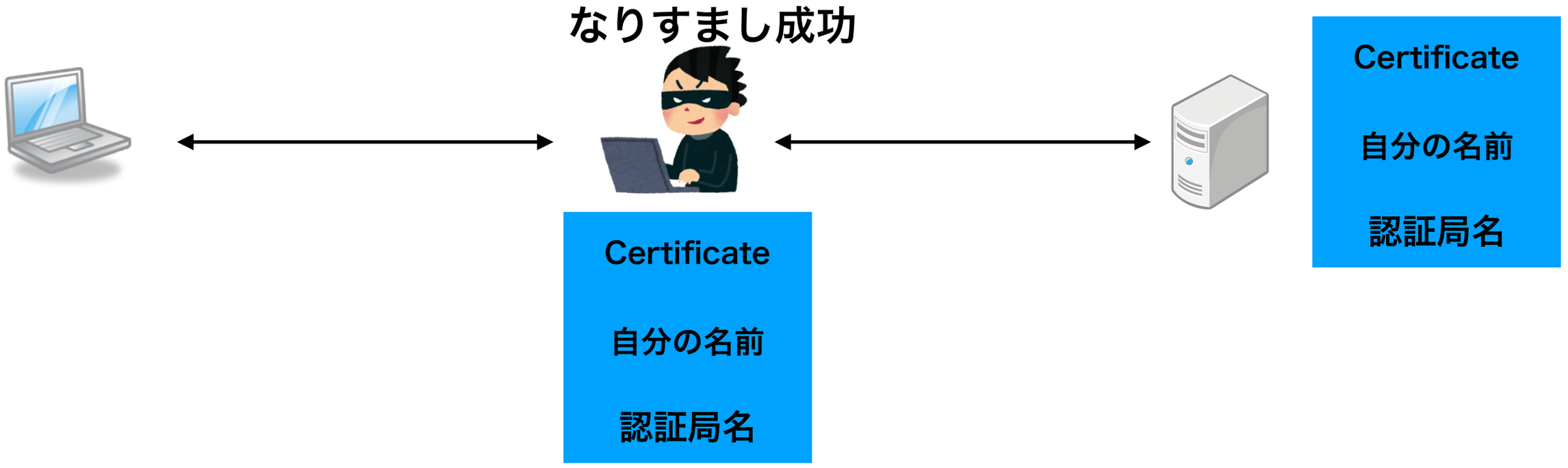
攻撃者による証明書の取得

攻撃者はサーバと個別にハンドシェイクをしてCertificateを入手します。



なりすまし攻撃

攻撃者は、サーバから事前に取得した Certificate を使ってなりすましを行いハンドシェイクをしましょう。

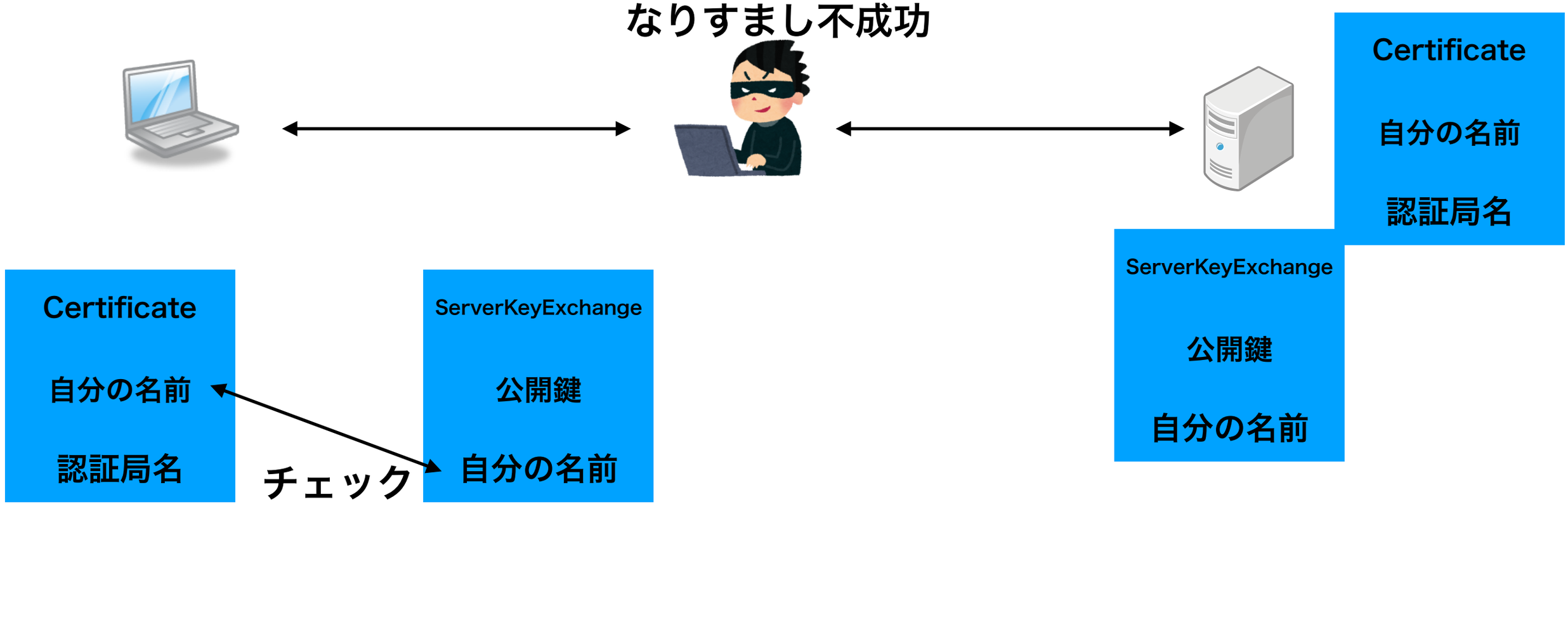


証明書は一般に公開されているものなので誰でも入手できます。

TLSなりすまし演習4
対策編
電子署名を使う

ServerKeyExchangeに署名

サーバ役の人は、ServerKeyExchangeに自分の名前を書いて署名しましょう。
クライアント役の人は、ServerKeyExchangeの署名がCertificateの名前と同一署名か確認してチェックしましょう。



TLSの認証まとめ

- ルート証明書によるサーバ証明書に対する署名
- ServerKeyExchangeに対するサーバの署名
- 上記２つのステップでなりすましを防ぐ

TLSハンドシェイクまとめ

- インターネット上の顔の見えない通信相手に対していかに安全な通信を確立するかという仕組み。以下のステップで行われる。

1. TLS通信を行う各種情報の交換
2. 鍵交換による共通鍵の生成
3. PKIを使った証明書と署名による通信相手の認証
4. ハンドシェイクの改ざん検知