

It Just (Net)works

The Truth About iOS'
Multipeer Connectivity Framework

Alban Diquet

@nabla_c0d3



About me

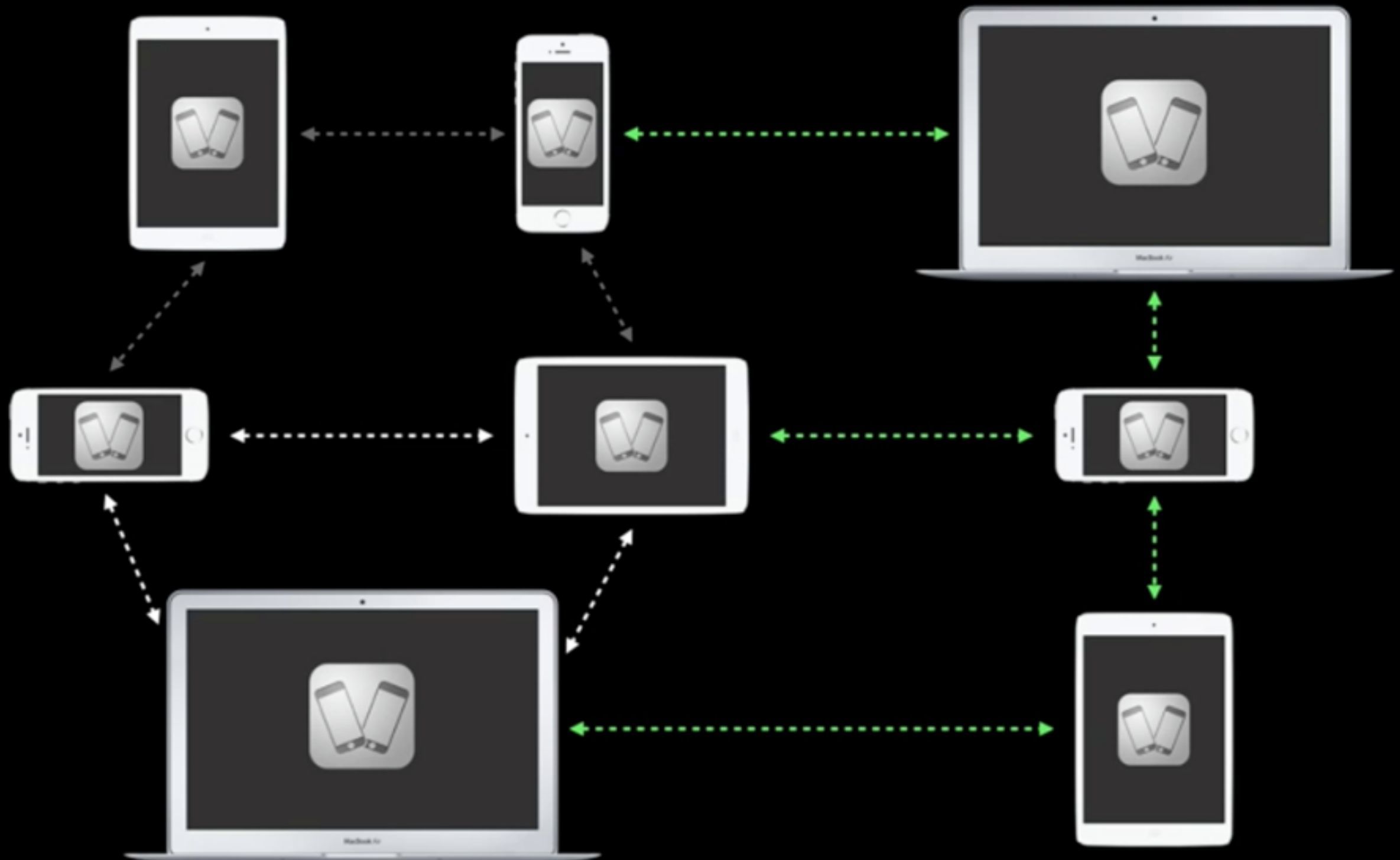
- iOS Security Researcher at Data Theorem
- Before: Principal Security Consultant at iSEC Partners
- Tools: SSLyze, Introspy, iOS SSL Kill Switch

Agenda

- What is Multipeer Connectivity?
- Reversing the MC protocol(s)
- Security analysis of MC

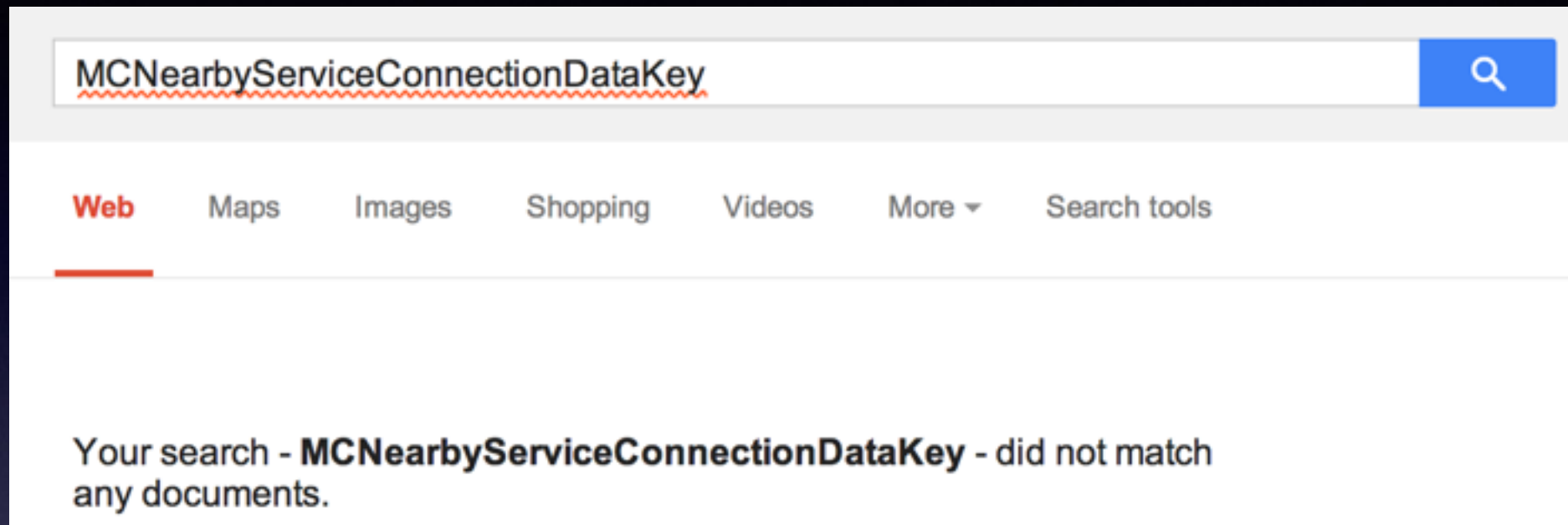
What is Multipeer Connectivity?

Multipeer Connectivity



Demo

Motivation



Encryption of session in MultipeerConnectivity framework for iOS



I am working on iOS multipeer framework and i am pretty happy with it. I am sharing some sensitive data so have to do the encryption. When we create the session we get three options `self.session = [[MCSession alloc] initWithPeer:self.myPeerID securityIdentity:nil encryptionPreference:MCEncryptionRequired];`

1. MCEncryptionNone
2. MCEncryptionOptional
3. MCEncryptionRequired

I read the Apple guide but couldn't find much info about it. If i pass MCEncryptionRequired, Does someone know what kind of encryption it does? Thanks

ios objective-c ipad multipeer-connectivity

Reversing the MC protocol(s)

MC API - Encryption

- The App can specify an *encryptionPreference*

```
– initWithPeer:securityIdentity:encryptionPreference:
```

- Three encryption levels:

`MCEncryptionOptional`

The session prefers to use encryption, but will accept unencrypted connections.

`MCEncryptionRequired`

The session requires encryption.

`MCEncryptionNone`

The session should not be encrypted.

- No further explanation in the documentation

MC API - Authentication

- The App can specify a *securityIdentity*

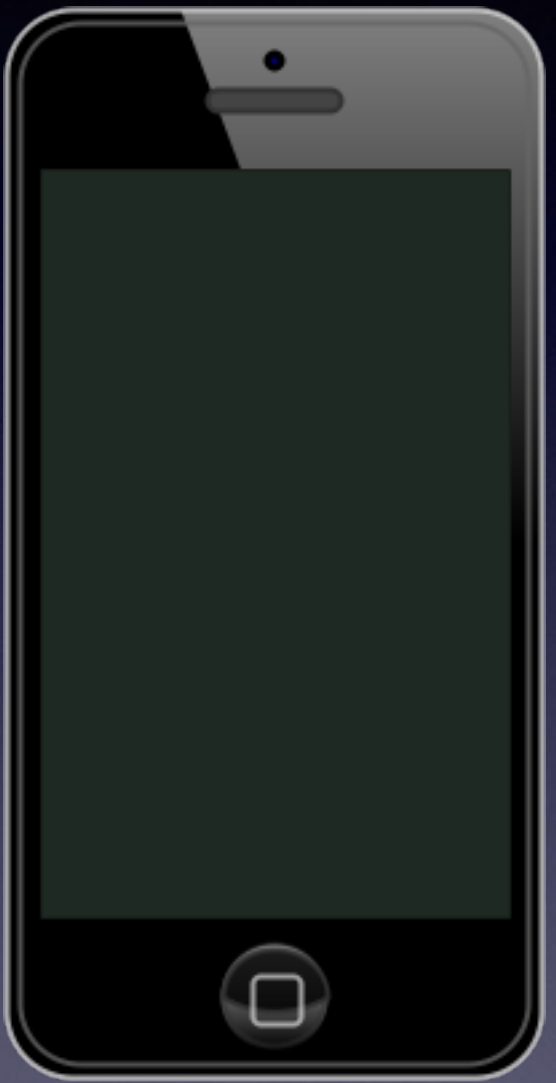
```
– initWithPeer:securityIdentity:encryptionPreference:
```

- A "security identity" is an X509 certificate and the corresponding private key
 - The peer's identify when pairing with other peers
- A callback has to be implemented for validating other peers' certificates/identities during pairing:

```
– session:didReceiveCertificate:fromPeer:certificateHandler:
```

Test Setup

- Macbook in WiFi Access Point mode + Wireshark
- Sample MC App with default MC settings
- Two devices:
 - iPad Air with Bluetooth disabled
 - iOS Simulator

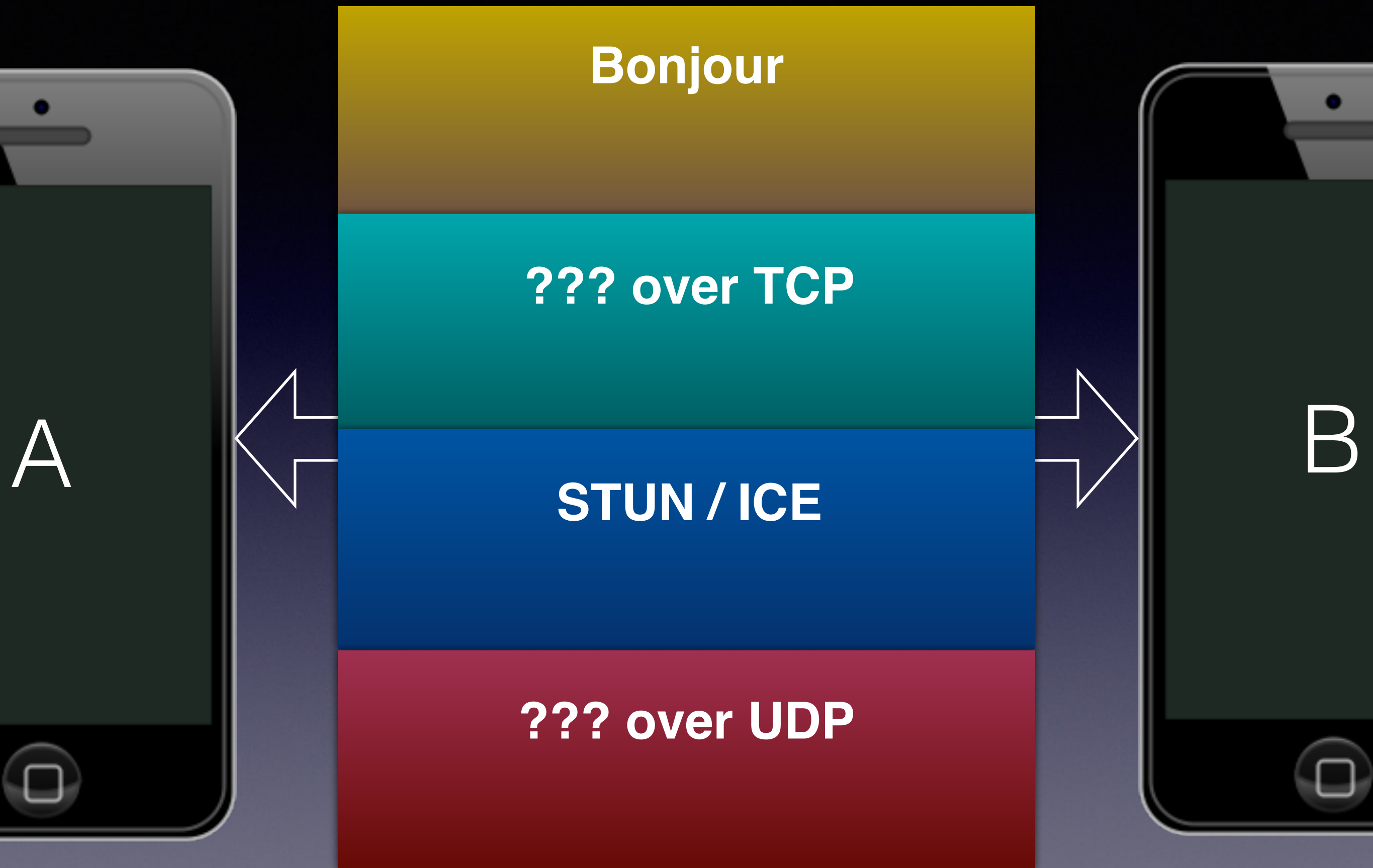




A

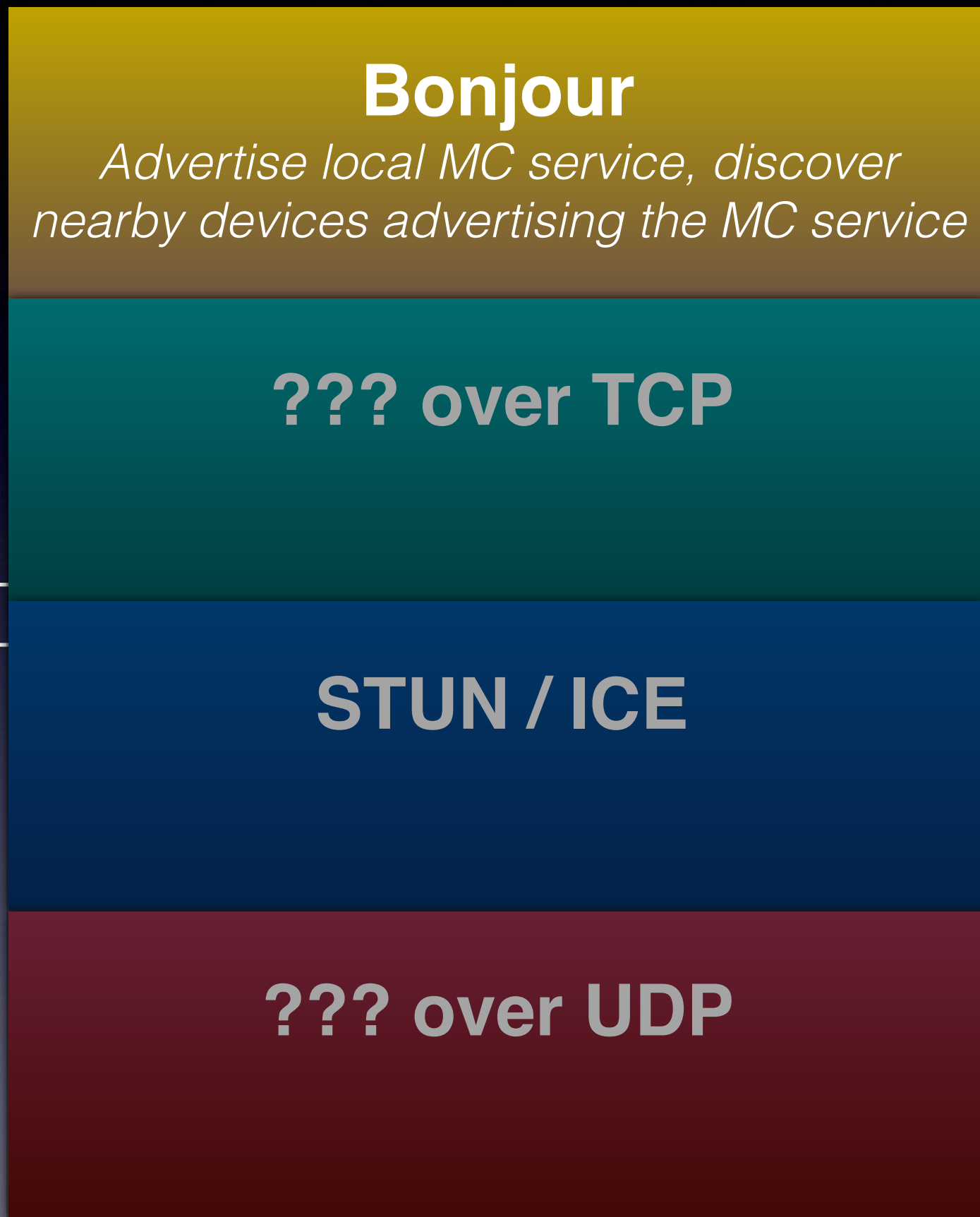
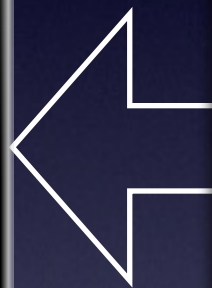
Source	Destination	Protocol	Length	Info
S3cr3t-iP...	224.0.0.2...	MDNS	185	Standard query...
S3cr3t-iP...	224.0.0.2...	MDNS	185	Standard query...
S3cr3t-iP...	224.0.0.2...	MDNS	442	Standard query...
S3cr3t-iP...	224.0.0.2...	MDNS	139	Standard query...
S3cr3t-iP...	224.0.0.2...	MDNS	442	Standard query...
192.168.1...	S3cr3t-iP...	TCP	440	51118 → 49585 ...
S3cr3t-iP...	192.168.1...	TCP	66	49585 → 51118 ...
S3cr3t-iP...	192.168.1...	TCP	82	49585 → 51118 ...
192.168.1...	S3cr3t-iP...	TCP	66	51118 → 49585 ...
S3cr3t-iP...	192.168.1...	TCP	464	49585 → 51118 ...
192.168.1...	S3cr3t-iP...	TCP	66	51118 → 49585 ...
192.168.1...	S3cr3t-iP...	TCP	82	51118 → 49585 ...
192.168.1...	S3cr3t-iP...	STUN	122	Binding Reques...
S3cr3t-iP...	192.168.1...	TCP	66	49585 → 51118 ...
S3cr3t-iP...	192.168.1...	STUN	122	Binding Reques...
192.168.1...	S3cr3t-iP...	STUN	130	Binding Succes...
S3cr3t-iP...	192.168.1...	STUN	130	Binding Succes...
S3cr3t-iP...	192.168.1...	STUN	134	Binding Reques...
192.168.1...	S3cr3t-iP...	STUN	130	Binding Succes...
S3cr3t-iP...	192.168.1...	UDP	118	Source port: 1...
192.168.1...	S3cr3t-iP...	UDP	138	Source port: 1...
192.168.1...	S3cr3t-iP...	UDP	843	Source port: 1...
192.168.1...	S3cr3t-iP...	UDP	68	Source port: 1...
192.168.1...	S3cr3t-iP...	UDP	138	Source port: 1...
192.168.1...	S3cr3t-iP...	UDP	843	Source port: 1...
192.168.1...	S3cr3t-iP...	UDP	68	Source port: 1...
S3cr3t-iP...	192.168.1...	UDP	326	Source port: 1...
S3cr3t-iP...	192.168.1...	UDP	60	Source port: 1...

B



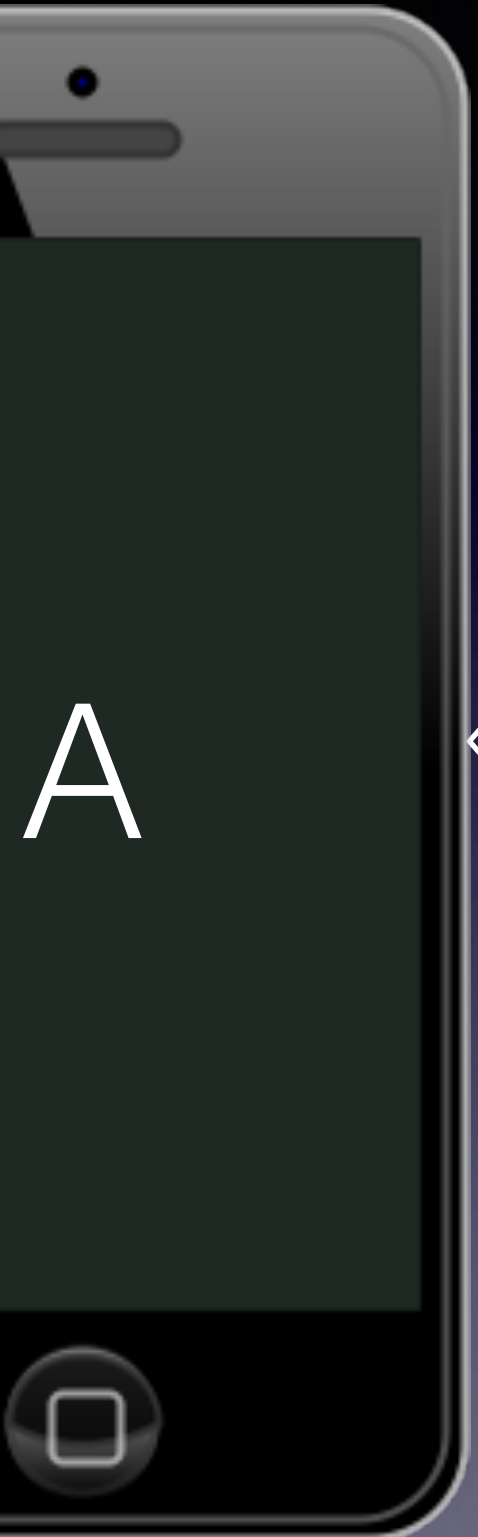


A



B





Bonjour

Advertise local MC service, discover nearby devices advertising the MC service

??? over TCP

STUN / ICE

??? over UDP





```
00000000 07 d0 00 00 00 00 00 25 a8 43 58 93 00 00 00 00 .....% .CX.....
00000010 00 00 00 06 00 1f 30 72 38 38 67 72 7a 76 63 71 .....0r 88grzvcq
00000020 65 70 65 2b 69 50 68 6f 6e 65 20 53 69 6d 75 6c epe+iPho ne Simul
00000030 61 74 6f 72 00 ator.
00000000 07 d0 00 00 00 00 00 20 49 cd 68 0a 00 00 00 00 ..... I.h.....
00000010 00 00 00 06 00 1a 33 6b 34 77 32 75 69 64 6d 76 .....3k 4w2uidmv
00000020 76 79 78 2b 53 33 63 72 33 74 20 69 50 61 64 00 vyx+S3cr 3t iPad.
00000030 07 d0 00 01 00 00 00 00 0c ca 7e 2c 00 00 00 00 ..... ..~,....
00000035 07 d0 00 01 00 00 00 00 0c ca 7e 2c 00 00 00 00 ..... ..~,....
00000045 08 98 00 00 00 00 00 00 2d c3 47 ff 00 00 00 01 ..... -.G.....
00000040 08 98 00 01 00 00 00 00 f0 55 9e 7a 00 00 00 01 ..... .U.z....
00000050 08 34 00 00 00 00 00 f1 86 bb 03 00 00 00 00 01 .4.....
00000060 62 70 6c 69 73 74 30 30 d4 01 02 03 04 05 06 07 bplist00 .....
00000070 08 5f 10 1a 4d 43 4e 65 61 72 62 79 53 65 72 76 .._.MCNe arbyServ
00000080 69 63 65 49 6e 76 69 74 65 49 44 4b 65 79 5f 10 iceInvit eIDKey_.
00000090 21 4d 43 4e 65 61 72 62 79 53 65 72 76 69 63 65 !MCNearb yService
000000A0 52 65 63 69 70 69 65 6e 74 50 65 65 72 49 44 4b Recipien tPeerIDK
000000B0 65 79 5f 10 1b 4d 43 4e 65 61 72 62 79 53 65 72 ey_..MCN earbySer
000000C0 76 69 63 65 4d 65 73 73 61 67 65 49 44 4b 65 79 viceMess ageIDKey
000000D0 5f 10 1e 4d 43 4e 65 61 72 62 79 53 65 72 76 69 _..MCNea rbyServi
000000E0 63 65 53 65 6e 64 65 72 50 65 65 72 49 44 4b 65 ceSender PeerIDKe
000000F0 79 10 00 4f 10 19 31 bc 8d 96 de 00 24 f2 10 69 y..0..1. ....$..i
00000100 50 68 6f 6e 65 20 53 69 6d 75 6c 61 74 6f 72 10 Phone Si mulator.
00000110 01 4f 10 14 ea 0e 27 21 05 e1 7d 99 0b 53 33 63 .0....'! ..}..S3c
00000120 72 33 74 20 69 50 61 64 08 11 2e 52 70 91 93 af r3t iPad ...Rp...
00000130 b1 00 00 00 00 00 00 01 01 00 00 00 00 00 00 00 .....
00000140 09 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000150 c8 .
00000055 08 34 00 01 00 00 00 00 73 e2 f9 bb 00 00 00 01 .4..... s.....
00000065 08 34 00 00 00 00 01 67 08 26 9f a3 00 00 00 02 .4.....g .&.....
00000075 62 70 6c 69 73 74 30 30 d6 01 02 03 04 05 06 07 bplist00 .....
00000085 08 09 0a 0b 0c 5f 10 20 4d 43 4e 65 61 72 62 79 ....._ MCNearby
```

6 client pkts, 5 server pkts, 6 turns.

Follow TCP Stream (tcp.stream eq 2)

00000000	07 d0 00 00 00 00 00 25	a8 43 58 93 00 00 00 00	% .CX.....
00000010	00 00 00 06 00 1f 30 72	38 38 67 72 7a 76 63 71	0r 88grzvcq
00000020	65 70 65 2b 69 50 68 6f	6e 65 20 53 69 6d 75 6c	epe+iPho ne Simul
00000030	61 74 6f 72 00		ator.
00000000	07 d0 00 00 00 00 00 20	49 cd 68 0a 00 00 00 00	 I.h.....
00000010	00 00 00 06 00 1a 33 6b	34 77 32 75 69 64 6d 76	3k 4w2uidmv
00000020	76 79 78 2b 53 33 63 72	33 74 20 69 50 61 64 00	vyx+S3cr 3t iPad.
00000030	07 d0 00 01 00 00 00 00	0c ca 7e 2c 00 00 00 00	~,....
00000035	07 d0 00 01 00 00 00 00	0c ca 7e 2c 00 00 00 00	~,....
00000045	08 98 00 00 00 00 00 00	2d c3 47 ff 00 00 00 01	 -.G.....
00000040	08 98 00 01 00 00 00 00	f0 55 9e 7a 00 00 00 01	U.z....
00000050	08 34 00 00 00 00 00 f1	86 bb 03 00 00 00 00 01	 4.....
00000060	62 70 6c 69 73 74 30 30	d4 01 02 03 04 05 06 07	bplist00.....
00000070	08 5f 10 1a 4d 43 4e 65	61 72 62 79 53 65 72 76	...MCNe arbyServ
00000080	69 63 65 49 6e 76 69 74	65 49 44 4b 65 79 5f 10	iceInvit eIDKey_.
00000090	21 4d 43 4e 65 61 72 62	79 53 65 72 76 69 63 65	!MCNearb yService
000000A0	52 65 63 69 70 69 65 6e	74 50 65 65 72 49 44 4b	Recipien tPeerIDK
000000B0	65 79 5f 10 1b 4d 43 4e	65 61 72 62 79 53 65 72	ey_..MCN earbySer
000000C0	76 69 63 65 4d 65 73 73	61 67 65 49 44 4b 65 79	viceMess ageIDKey
000000D0	5f 10 1e 4d 43 4e 65 61	72 62 79 53 65 72 76 69	_..MCNea rbyServi
000000E0	63 65 53 65 6e 64 65 72	50 65 65 72 49 44 4b 65	ceSender PeerIDKe
000000F0	79 10 00 4f 10 19 31 bc	8d 96 de 00 24 f2 10 69	y..0..1.\$.i
00000100	50 68 6f 6e 65 20 53 69	6d 75 6c 61 74 6f 72 10	Phone Si mulator.
00000110	01 4f 10 14 ea 0e 27 21	05 e1 7d 99 0b 53 33 63	.0....'! ..}..S3c
00000120	72 33 74 20 69 50 61 64	08 11 2e 52 70 91 93 af	r3t iPad ...Rp...
00000130	b1 00 00 00 00 00 00 01	01 00 00 00 00 00 00 00	
00000140	09 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000150	c8		.
00000055	08 34 00 01 00 00 00 00	73 e2 f9 bb 00 00 00 01	.4..... s.....
00000065	08 34 00 00 00 00 01 67	08 26 9f a3 00 00 00 02	.4.....g &.....
00000075	62 70 6c 69 73 74 30 30	d6 01 02 03 04 05 06 07	bplist00.....
00000085	08 09 0a 0b 0c 5f 10 20	4d 43 4e 65 61 72 62 79	_.. MCNearby

6 client pkts, 5 server pkts, 6 turns.

Mystery Protocol #1

- Peer connects to the other peer over TCP
- Each peer sends their “PeerID” first
 - (random) “idString” + device name
 - For example: “ory2g6r8fkq+iPhone Simulator”
- Three plists are then exchanged



Key	Type	Value
▼ Root	Dictionary	(4 items)
MCNearbyServiceInviteIDKey	Number	0
MCNearbyServiceRecipientPeerIDKey	Data	<0141d383 ecf553da 10695068
MCNearbyServiceMessageIDKey	Number	1
MCNearbyServiceSenderPeerIDKey	Data	<d9de1832 f26fedf7 0b533363 7





Key	Type	Value
▼ Root	Dictionary	(4 items)
MCNearbyServiceInviteIDKey	Number	0
MCNearbyServiceRecipientPeerIDKey	Data	<0141d383 ecf553da 10695068
MCNearbyServiceMessageIDKey	Number	1
MCNearbyServiceSenderPeerIDKey	Data	<d9de1832 f26fedf7 0b533363 7



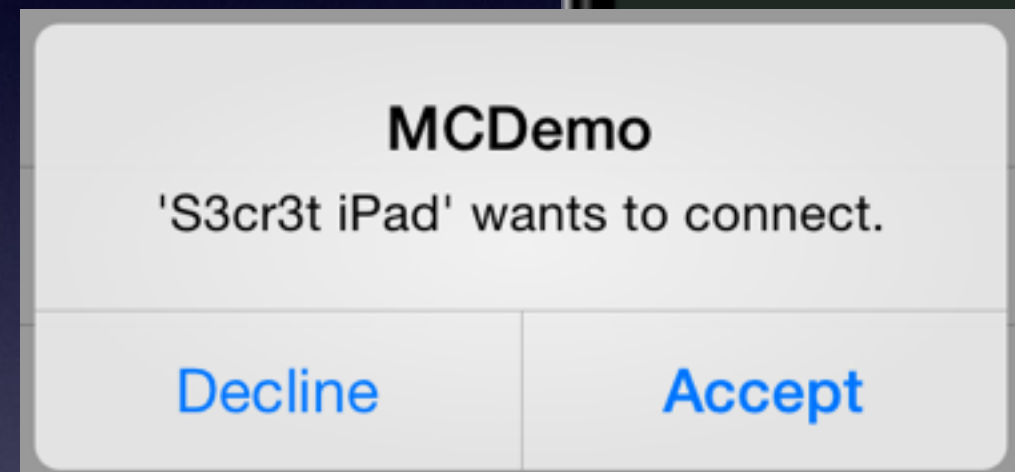


Key	Type	Value
▼ Root	Dictionary	(4 items)
MCNearbyServiceInviteIDKey	Number	0
MCNearbyServiceRecipientPeer...	Data	<0141d383 ecf553da 10695068
MCNearbyServiceMessageIDKey	Number	1
MCNearbyServiceSenderPeerIDKey	Data	<d9de1832 f26fedf7 0b533363 7



A

Key	Type	Value
▼ Root	Dictionary	(4 items)
MCNearbyServiceInviteIDKey	Number	0
MCNearbyServiceRecipientPeerIDKey	Data	<0141d383 ecf553da 10695068
MCNearbyServiceMessageIDKey	Number	1
MCNearbyServiceSenderPeerIDKey	Data	<d9de1832 f26fedf7 0b533363 7





Key	Type	Value
▼ Root	Dictionary	(4 items)
MCNearbyServiceInviteIDKey	Number	0
MCNearbyServiceRecipientPeer...	Data	<0141d383 ecf553da 10695068
MCNearbyServiceMessageIDKey	Number	1
MCNearbyServiceSenderPeerIDKey	Data	<d9de1832 f26fedf7 0b533363 7

Key	Type	Value
▼ Root	Dictionary	(6 items)
MCNearbyServiceSenderPeerIDKey	Data	<0141d383 ecf553da 10695068
MCNearbyServiceInviteIDKey	Number	0
MCNearbyServiceAcceptInviteKey	Boolean	YES
MCNearbyServiceConnectionDataKey	Data	<80000019 100501a8 c061f7ed
MCNearbyServiceRecipientPeerIDKey	Data	<d9de1832 f26fedf7 0b533363 7
MCNearbyServiceMessageIDKey	Number	2





Key	Type	Value
▼ Root	Dictionary	(4 items)
MCNearbyServiceInviteIDKey	Number	0
MCNearbyServiceRecipientPeer...	Data	<0141d383 ecf553da 10695068
MCNearbyServiceMessageIDKey	Number	1
MCNearbyServiceSenderPeerIDKey	Data	<d9de1832 f26fedf7 0b533363 7

Key	Type	Value
▼ Root	Dictionary	(6 items)
MCNearbyServiceSenderPeerIDKey	Data	<0141d383 ecf553da 10695068
MCNearbyServiceInviteIDKey	Number	0
MCNearbyServiceAcceptInviteKey	Boolean	YES
MCNearbyServiceConnectionDataKey	Data	<80000019 100501a8 c061f7ed
MCNearbyServiceRecipientPeerIDKey	Data	<d9de1832 f26fedf7 0b533363 7
MCNearbyServiceMessageIDKey	Number	2



A

Key	Type	Value
▼ Root	Dictionary	(4 items)
MCNearbyServiceInviteIDKey	Number	0
MCNearbyServiceRecipientPeerIDKey	Data	<0141d383 ecf553da 10695068
MCNearbyServiceMessageIDKey	Number	1
MCNearbyServiceSenderPeerIDKey	Data	<d9de1832 f26fedf7 0b533363 7

Key	Type	Value
▼ Root	Dictionary	(6 items)
MCNearbyServiceSenderPeerIDKey	Data	<0141d383 ecf553da 10695068
MCNearbyServiceInviteIDKey	Number	0
MCNearbyServiceAcceptInviteKey	Boolean	YES
MCNearbyServiceConnectionDataKey	Data	<80000019 100501a8 c061f7ed
MCNearbyServiceRecipientPeerIDKey	Data	<d9de1832 f26fedf7 0b533363 7
MCNearbyServiceMessageIDKey	Number	2

Key	Type	Value
▼ Root	Dictionary	(5 items)
MCNearbyServiceSenderPeerIDKey	Data	<d9de1832 f26fedf7 0b533363 7
MCNearbyServiceInviteIDKey	Number	0
MCNearbyServiceConnectionDataKey	Data	<80000059 120f01a8 c0fe8000 0
MCNearbyServiceRecipientPeerIDKey	Data	<0141d383 ecf553da 10695068
MCNearbyServiceMessageIDKey	Number	3

B

A

Key	Type	Value
▼ Root	Dictionary	(4 items)
MCNearbyServiceInviteIDKey	Number	0
MCNearbyServiceRecipientPeerIDKey	Data	<0141d383 ecf553da 10695068
MCNearbyServiceMessageIDKey	Number	1
MCNearbyServiceSenderPeerIDKey	Data	<d9de1832 f26fedf7 0b533363 7

Key	Type	Value
▼ Root	Dictionary	(6 items)
MCNearbyServiceSenderPeerIDKey	Data	<0141d383 ecf553da 10695068
MCNearbyServiceInviteIDKey	Number	0
MCNearbyServiceAcceptInviteKey	Boolean	YES
MCNearbyServiceConnectionDataKey	Data	<80000019 100501a8 c061f7ed
MCNearbyServiceRecipientPeerIDKey	Data	<d9de1832 f26fedf7 0b533363 7
MCNearbyServiceMessageIDKey	Number	2

Key	Type	Value
▼ Root	Dictionary	(5 items)
MCNearbyServiceSenderPeerIDKey	Data	<d9de1832 f26fedf7 0b533363 7
MCNearbyServiceInviteIDKey	Number	0
MCNearbyServiceConnectionDataKey	Data	<80000059 120f01a8 c0fe8000 0
MCNearbyServiceRecipientPeerIDKey	Data	<0141d383 ecf553da 10695068
MCNearbyServiceMessageIDKey	Number	3

B

Mystery Protocol #1

- Each peer exchanges their **MCNearbyConnectionDataKey**
 - Main "payload" of the protocol; briefly mentioned as "connection data" in the documentation

```
80050041 300801A8 C069EAFE A90102A8 C0611237 7F506F7D 4FE35A00 00008011
40611237 7F506474 62125A00 00008111 40611237 7F5045A8 7A145A00 00008211
40
```


Mystery Protocol #1

- Each peer exchanges their **MCNearbyConnectionDataKey**
 - Main "payload" of the protocol; briefly mentioned as "connection data" in the documentation

```
80050041 300801A8 C069EAFE A90102A8 C0611237 7F506F7D 4FE35A00 00008011
40611237 7F506474 62125A00 00008111 40611237 7F5045A8 7A145A00 00008211
40
```

- The peer's security settings as bit fields:
 - Encryption level (optional = X00, none = X10, required = X01)
 - Whether authentication is enabled (yes = 1XX, no = 0XX)
 - No X509 certificate/identity yet

Mystery Protocol #1

- Each peer exchanges their **MCNearbyConnectionDataKey**
 - Main "payload" of the protocol; briefly mentioned as "connection data" in the documentation

```
80050041 300801A8 C069EAFE A90102A8 C0611237 7F506F7D 4FE35A00 00008011
40611237 7F506474 62125A00 00008111 40611237 7F5045A8 7A145A00 00008211
40
```

- Then a list of local "candidate" IP addresses

Mystery Protocol #1

- Each peer exchanges their **MCNearbyConnectionDataKey**
 - Main "payload" of the protocol; briefly mentioned as "connection data" in the documentation

```
80050041 300801A8 C069EAFE A90102A8 C0611237 7F506F7D 4FE35A00 00008011
40611237 7F506474 62125A00 00008111 40611237 7F5045A8 7A145A00 00008211
40
```

- Then a list of local "candidate" IP addresses
 - 192.168.1.8

Mystery Protocol #1

- Each peer exchanges their **MCNearbyConnectionDataKey**
 - Main "payload" of the protocol; briefly mentioned as "connection data" in the documentation

```
80050041 300801A8 C069EAFE A90102A8 C0611237 7F506F7D 4FE35A00 00008011
40611237 7F506474 62125A00 00008111 40611237 7F5045A8 7A145A00 00008211
40
```

- Then a list of local "candidate" IP addresses
 - 192.168.1.8
 - 169.254.234.105
 - Etc...

Mystery Protocol #1

- Each peer exchanges their **MCNearbyConnectionDataKey**
 - Main "payload" of the protocol; briefly mentioned as "connection data" in the documentation

```
80050041 300801A8 C069EAFE A90102A8 C0611237 7F506F7D 4FE35A00 00008011
40611237 7F506474 62125A00 00008111 40611237 7F5045A8 7A145A00 00008211
40
```

- Then some kind of IDs (according to debug logs)?

Mystery Protocol #1

- Each peer exchanges their **MCNearbyConnectionDataKey**
- Main "payload" of the protocol; briefly mentioned as "connection data" in the documentation

```
80050041 300801A8 C069EAFE A90102A8 C0611237 7F506F7D 4FE35A00 00008011
40611237 7F506474 62125A00 00008111 40611237 7F5045A8 7A145A00 00008211
40
```

- Then some kind of IDs (according to debug logs)?
- 6F7D4FE3, etc...

```
ID[6F7D4FE300000000] [192.168.1.8:16401] flag(08).
ID[6474621200000000] [169.254.234.105:16401] flag(08).
ID[45A87A1400000000] [192.168.2.1:16401] flag(08).
```

A

Bonjour

Advertise local MC service, discover nearby devices advertising the MC service

GCK1 over TCP

Exchange peer names, security options and "candidate" UDP sockets

STUN / ICE

??? over UDP

B

A

Bonjour

Advertise local MC service, discover nearby devices advertising the MC service

GCK1 over TCP

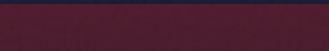
Exchange peer names, security options and "candidate" UDP sockets

STUN / ICE

??? over UDP

B

Interactive Connectivity Establishment



```
com.apple.ICE: Local candidate(1/3): ID[07FEE53F00000000] [192.168.2.2:16402] flag(08) if(en0).
com.apple.ICE: Local candidate(2/3): ID[4348FA0000000000] [[fe80::29:203:1454:aa5a%en0]:16402]
com.apple.ICE: Local candidate(3/3): ID[3904EA8D00000000] [[fe80::ecf1:14ff:fe49:d55a%awdl0]:16402]
com.apple.ICE: Remote candidate(1/3): ID[6F7D4FE300000000] [192.168.1.8:16401] flag(08).
com.apple.ICE: Remote candidate(2/3): ID[6474621200000000] [169.254.234.105:16401] flag(08).
com.apple.ICE: Remote candidate(3/3): ID[45A87A1400000000] [192.168.2.1:16401] flag(08).
```

Bonjour

Advertise local MC service, discover nearby devices advertising the MC service

GCK1 over TCP

Exchange peer names, security options and "candidate" UDP sockets

STUN / ICE

Perform connectivity checks and find the best network path to the other peer

??? over UDP

A

B

Bonjour

Advertise local MC service, discover nearby devices advertising the MC service

GCK1 over TCP

Exchange peer names, security options and "candidate" UDP sockets

STUN / ICE

Perform connectivity checks and find the best network path to the other peer

??? over UDP

A

B

Mystery Protocol #2

Source	Destination	Protocol	Length	Info
192.168.1....	MacBook-Pr...	UDP	190	Source port: 16402 Destination port: ...
MacBook-Pr...	192.168.1....	UDP	138	Source port: 16402 Destination port: ...
MacBook-Pr...	192.168.1....	UDP	856	Source port: 16402 Destination port: ...
MacBook-Pr...	192.168.1....	UDP	73	Source port: 16402 Destination port: ...
MacBook-Pr...	192.168.1....	UDP	68	Source port: 16402 Destination port: ...
192.168.1....	MacBook-Pr...	UDP	856	Source port: 16402 Destination port: ...
192.168.1....	MacBook-Pr...	UDP	198	Source port: 16402 Destination port: ...
192.168.1....	MacBook-Pr...	UDP	198	Source port: 16402 Destination port: ...
192.168.1....	MacBook-Pr...	UDP	60	Source port: 16402 Destination port: ...
192.168.1....	MacBook-Pr...	UDP	136	Source port: 16402 Destination port: ...
MacBook-Pr...	192.168.1....	UDP	57	Source port: 16402 Destination port: ...
MacBook-Pr...	192.168.1....	UDP	136	Source port: 16402 Destination port: ...

Mystery Protocol #2

Follow UDP Stream ((ip.addr eq 192.168.1.15 and ip.addr eq 192.168.1.5) and (udp.port eq...

000004F8	d0 14 fe ff 00 00 00 00	00 00 00 04 00 01 01	
00000507	d0 16 fe ff 00 01 00 00	00 00 00 01 00 40 94 53	@.S
00000517	45 76 f9 0a 37 4f 03 67	5f 8d 2d 54 14 12 65 a4	Ev..70.g _.-T..e.
00000527	b8 ec 86 76 b9 4c 25 dc	2a 63 9d 58 74 aa e1 ce	...v.L%. *c.Xt...
00000537	75 7a 3d c5 20 15 c0 91	8a 57 3d 6a 1f a8 8b 7c	uz=.W=j...
00000547	ae da fd e2 88 72 2b 2a	4a 7d a1 28 20 87	r+* J}.(.
0000049E	d0 17 fe ff 00 01 00 00	00 00 00 03 00 50 70 94	Pp.
000004AE	48 9f 70 cb d5 42 78 17	af 3a 94 78 01 37 37 0a	H.p..Bx. ...x.77.
000004BE	3a 61 49 91 a3 3f 66 9f	0e e1 f8 45 34 6e e0 64	:aI..?f. ...E4n.d
000004CE	1f 4f f9 88 97 64 e4 dc	dc 30 d6 7e aa 1d d2 88	.0...d.. .0.~....
000004DE	6a fd d1 f0 bd a2 03 63	8f cb 1f e9 66 c2 7d 74	j.....cf.}t
000004EE	2c 79 42 27 61 ae 9e 7a	cc 09 ef 75 0c 17	,yB'a..z ...u..
00000555	d0 14 fe ff 00 00 00 00	00 00 00 05 00 01 01	
00000564	d0 16 fe ff 00 01 00 00	00 00 00 02 00 40 93 8f	@..
00000574	f3 6c 59 a7 e0 8d 55 89	f8 93 9f b9 3c 79 2e 41	.lY...U.<y.A
00000584	4b 59 01 10 45 bf 84 c7	2c d0 60 dd f6 d4 66 5b	KY..E... ,.`...f[
00000594	6b 48 31 16 e0 36 cf af	65 58 7d 1d 58 11 15 09	kH1..6.. eX}.X...
000005A4	c4 5f 33 4c d5 20 66 f3	d8 6c c4 0e fe 37	._3L. f. .l...7
000004FC	d0 17 fe ff 00 01 00 00	00 00 00 04 00 50 78 14	Px.
0000050C	1b 08 53 e8 b5 92 bc bf	3c 42 84 f6 11 c9 7d a6	..S..... <B....}.

Packet 179. 25 client pkts, 27 server pkts, 31 turns. Click to select.

Mystery Protocol #2

- It's the protocol used when App data is being exchanged
- Not plaintext... but Wireshark doesn't know what it is
- Clues:
 - Authentication in the MC API relies on X509 certificates
 -

Mystery Protocol #2

- It's the protocol used when App data is being exchanged
- Not plaintext... but Wireshark doesn't know what it is
- Clues:
 - Authentication in the MC API relies on X509 certificates
 - When setting a breakpoint on SSLHandshake(), it does get triggered...

```
(lldb) break set --name SSLHandshake
Breakpoint 1: where = Security`SSLHandshake, address = 0x31a3dc8c
(lldb) bt
* thread #8: tid = 0x6d513, 0x31a3dc8c Security`SSLHandshake, name =
'com.apple.gamekit.services.gcksession.recvproc', stop reason = breakpoint 1.1
  * frame #0: 0x31a3dc8c Security`SSLHandshake
    frame #1: 0x30c88bbe MultipeerConnectivity`gckSessionPerformDTLSHandshake + 134
    frame #2: 0x30c813fe MultipeerConnectivity`gckSessionRecvProc + 2718
    frame #3: 0x3a13dc5c libsystem_pthread.dylib`_pthread_body + 140
    frame #4: 0x3a13dbce libsystem_pthread.dylib`_pthread_start + 102
```

Mystery Protocol #2

- It's the protocol used when App data is being exchanged
- Not plaintext... but Wireshark doesn't know what it is
- Clues:
 - Authentication in the MC API relies on X509 certificates
 - When setting a breakpoint on SSLHandshake(), it does get triggered...

```
(lldb) break set --name SSLHandshake
Breakpoint 1: where = Security`SSLHandshake, address = 0x31a3dc8c
(lldb) bt
* thread #8: tid = 0x6d513, 0x31a3dc8c Security`SSLHandshake, name =
'com.apple.gamekit.services.gcksession.recvproc', stop reason = breakpoint 1.1
  * frame #0: 0x31a3dc8c Security`SSLHandshake
    frame #1: 0x30c88bbe MultipeerConnectivity`gckSessionPerformDTLSHandshake + 134
    frame #2: 0x30c813fe MultipeerConnectivity`gckSessionRecvProc + 27/18
    frame #3: 0x3a13dc5c libsystem_pthread.dylib`_pthread_body + 140
    frame #4: 0x3a13dbce libsystem_pthread.dylib`_pthread_start + 102
```


Mystery Protocol #2

```
Follow UDP Stream ((ip.addr eq 192.168.1.14 and ip.addr eq 192.168.1.13) and (udp.port e...

00000154 d0 16 fe ff 00 00 00 00 00 00 00 00 00 86 01 00 .....
00000164 00 7a 00 00 00 00 00 00 00 7a fe ff 53 84 24 71 .z..... .z..S.$q
00000174 b5 0c 5e 43 00 2c e9 25 21 e6 1c 2d 52 4c fe 28 ..^C.,.% !..-RL.(
00000184 81 59 38 04 58 68 56 44 0e 1e 44 1d 00 00 00 3e .Y8.XhVD ..D....>
00000194 00 ff c0 24 c0 23 c0 0a c0 09 c0 08 c0 28 c0 27 ...$.#.. ....('
000001A4 c0 14 c0 13 c0 12 c0 26 c0 25 c0 2a c0 29 c0 05 .....& .%.*.)..
000001B4 c0 04 c0 03 c0 0f c0 0e c0 0d 00 3d 00 3c 00 2f ..... ...=<./
000001C4 00 35 00 0a 00 67 00 6b 00 33 00 39 00 16 01 00 .5...g.k .3.9....
000001D4 00 12 00 0a 00 08 00 06 00 17 00 18 00 19 00 0b .....
000001E4 00 02 01 00 .....
000001F8 d0 16 fe ff 00 00 00 00 00 00 00 00 00 52 02 00 .....R..
00000208 00 46 00 00 00 00 00 00 00 46 fe ff 53 84 24 71 .F..... .F..S.$q
00000218 6a ee 06 ec 4b 73 3d 21 38 ef be a6 28 ee 75 98 j...Ks=! 8...(u.
```

openssl s_client -dtls1 -connect someserver:443

```
Follow UDP Stream ((ip.addr eq 127.0.0.1 and ip.addr eq 127.0.0.1) and (udp.port eq 49857...

0000006F 16 fe ff 00 00 00 00 00 00 00 01 00 76 01 00 00 .....v...
0000007F 6a 00 01 00 00 00 00 00 6a fe ff 53 b9 a1 1a a2 j..... j..S....
0000008F 81 82 ac 40 d1 fa db 74 f7 a3 03 71 46 e2 c9 83 ...@...t ...qF...
0000009F 38 46 4b 7c 4e 98 f8 60 03 f1 3f 00 14 ef f0 65 8FK|N..` ..?....e
000000AF a1 7f e7 9f cb c1 4d 0f b8 06 e5 2f 00 85 98 7c .....M. .../...|
000000BF 4c 00 28 00 39 00 38 00 35 00 16 00 13 00 0a 00 L.(.9.8. 5.....
000000CF 33 00 32 00 2f 00 9a 00 99 00 96 00 15 00 12 00 3.2./... .....
000000DF 09 00 14 00 11 00 08 00 06 00 ff 01 00 00 04 00 .....
000000EF 23 00 00 .....#..
00000030 16 fe ff 00 00 00 00 00 00 00 01 00 3d 02 00 00 .....=...
00000040 31 00 01 00 00 00 00 00 31 fe ff 53 b9 a1 1a a3 1..... 1..S....
00000050 6f 9d 49 e3 b5 7d cf 91 06 37 37 10 4b 79 15 80 o.I..}.. .77.Ky..
```


Mystery Protocol #2

```
Follow UDP Stream ((ip.addr eq 192.168.1.14 and ip.addr eq 192.168.1.13) and (udp.port e...

00000154 d0 16 fe ff 00 00 00 00 00 00 00 00 00 86 01 00 .....
00000164 00 7a 00 00 00 00 00 00 00 7a fe ff 53 84 24 71 .z..... .z..S.$q
00000174 b5 0c 5e 43 00 2c e9 25 21 e6 1c 2d 52 4c fe 28 ..^C.,.% !..-RL.(
00000184 81 59 38 04 58 68 56 44 0e 1e 44 1d 00 00 00 3e .Y8.XhVD ..D....>
00000194 00 ff c0 24 c0 23 c0 0a c0 09 c0 08 c0 28 c0 27 ...$.#.. ....('
000001A4 c0 14 c0 13 c0 12 c0 26 c0 25 c0 2a c0 29 c0 05 .....& .%.*.)..
000001B4 c0 04 c0 03 c0 0f c0 0e c0 0d 00 3d 00 3c 00 2f ..... ...=<./
000001C4 00 35 00 0a 00 67 00 6b 00 33 00 39 00 16 01 00 .5...g.k .3.9....
000001D4 00 12 00 0a 00 08 00 06 00 17 00 18 00 19 00 0b .....
000001E4 00 02 01 00 .....
000001F8 d0 16 fe ff 00 00 00 00 00 00 00 00 52 02 00 .....R..
00000208 00 46 00 00 00 00 00 00 00 46 fe ff 53 84 24 71 .F..... .F..S.$q
00000218 6a ee 06 ec 4b 73 3d 21 38 ef be a6 28 ee 75 98 j...Ks=! 8...(u.
```

`openssl s_client -dtls1 -connect someserver:443`

```
Follow UDP Stream ((ip.addr eq 127.0.0.1 and ip.addr eq 127.0.0.1) and (udp.port eq 49857...

0000006F 16 fe ff 00 00 00 00 00 00 00 01 00 76 01 00 00 .....v...
0000007F 6a 00 01 00 00 00 00 00 6a fe ff 53 b9 a1 1a a2 j..... j..S....
0000008F 81 82 ac 40 d1 fa db 74 f7 a3 03 71 46 e2 c9 83 ...@...t ...qF...
0000009F 38 46 4b 7c 4e 98 f8 60 03 f1 3f 00 14 ef f0 65 8FK|N..` ..?....e
000000AF a1 7f e7 9f cb c1 4d 0f b8 06 e5 2f 00 85 98 7c .....M. .../...|
000000BF 4c 00 28 00 39 00 38 00 35 00 16 00 13 00 0a 00 L.(.9.8. 5.....
000000CF 33 00 32 00 2f 00 9a 00 99 00 96 00 15 00 12 00 3.2./... .....
000000DF 09 00 14 00 11 00 08 00 06 00 ff 01 00 00 04 00 .....
000000EF 23 00 00 .....#..
00000030 16 fe ff 00 00 00 00 00 00 00 01 00 3d 02 00 00 .....=...
00000040 31 00 01 00 00 00 00 00 31 fe ff 53 b9 a1 1a a3 1..... 1..S....
00000050 6f 9d 49 e3 b5 7d cf 91 06 37 37 10 4b 79 15 80 o.I..}.. .77.Ky..
```


Mystery Protocol #2

```
Follow UDP Stream ((ip.addr eq 192.168.1.14 and ip.addr eq 192.168.1.13) and (udp.port e...

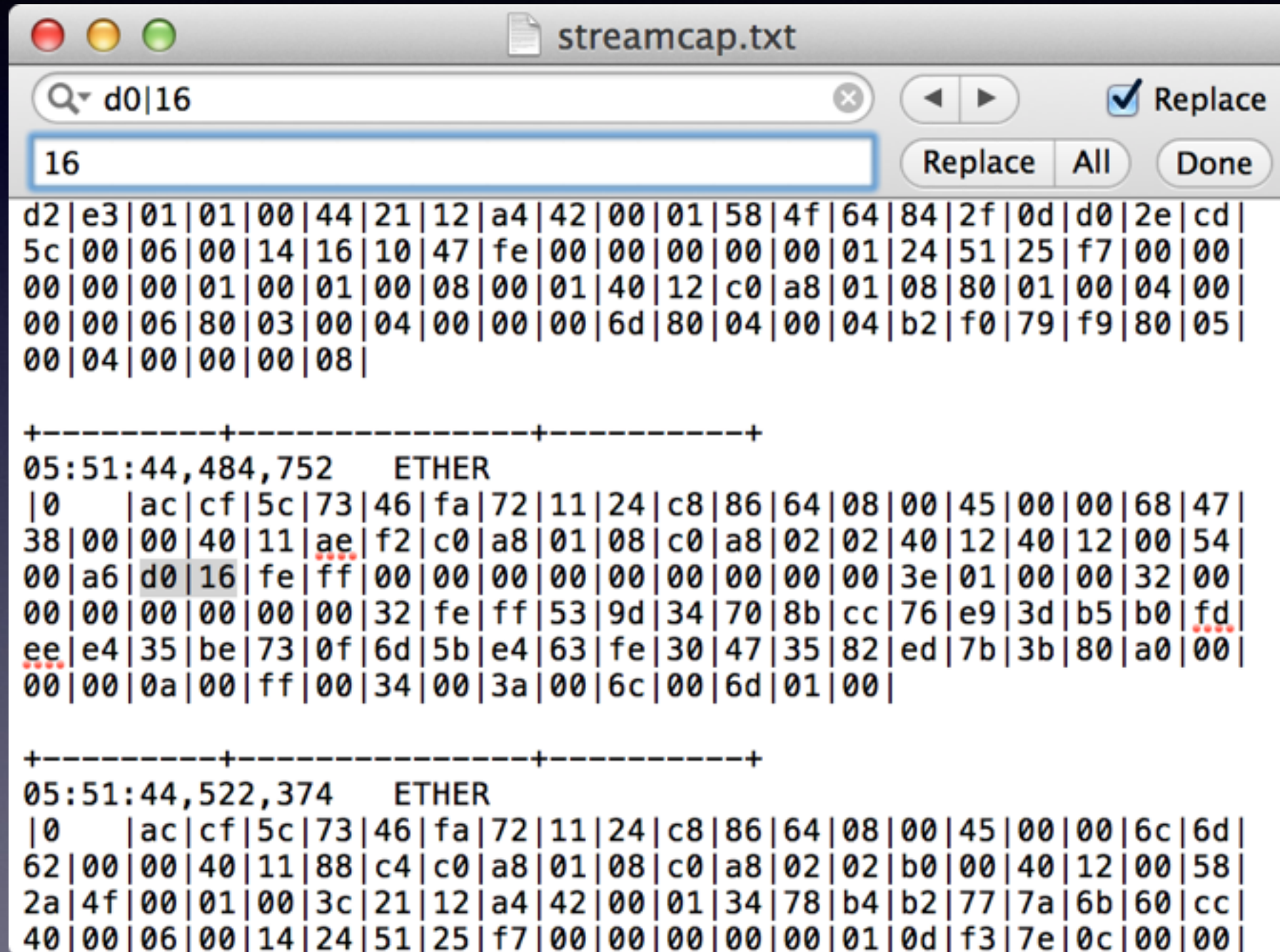
00000154 d0 16 fe ff 00 00 00 00 00 00 00 00 00 86 01 00 .....
00000164 00 7a 00 00 00 00 00 00 00 7a fe ff 53 84 24 71 .z..... .z..S.$q
00000174 b5 0c 5e 43 00 2c e9 25 21 e6 1c 2d 52 4c fe 28 ..^C.,.% !..-RL.(
00000184 81 59 38 04 58 68 56 44 0e 1e 44 1d 00 00 00 3e .Y8.XhVD ..D....>
00000194 00 ff c0 24 c0 23 c0 0a c0 09 c0 08 c0 28 c0 27 ...$.#.. ....('
000001A4 c0 14 c0 13 c0 12 c0 26 c0 25 c0 2a c0 29 c0 05 .....& .%.*.)..
000001B4 c0 04 c0 03 c0 0f c0 0e c0 0d 00 3d 00 3c 00 2f ..... ...=<./
000001C4 00 35 00 0a 00 67 00 6b 00 33 00 39 00 16 01 00 .5...g.k .3.9....
000001D4 00 12 00 0a 00 08 00 06 00 17 00 18 00 19 00 0b .....
000001E4 00 02 01 00 .....
000001F8 d0 16 fe ff 00 00 00 00 00 00 00 00 00 52 02 00 .....R..
00000208 00 46 00 00 00 00 00 00 00 46 fe ff 53 84 24 71 .F..... .F..S.$q
00000218 6a ee 06 ec 4b 73 3d 21 38 ef be a6 28 ee 75 98 j...Ks=! 8...(u.
```

`openssl s_client -dtls1 -connect someserver:443`

```
Follow UDP Stream ((ip.addr eq 127.0.0.1 and ip.addr eq 127.0.0.1) and (udp.port eq 49857...

0000006F 16 fe ff 00 00 00 00 00 00 00 01 00 76 01 00 00 .....v...
0000007F 6a 00 01 00 00 00 00 00 6a fe ff 53 b9 a1 1a a2 j..... j..S....
0000008F 81 82 ac 40 d1 fa db 74 f7 a3 03 71 46 e2 c9 83 ...@...t ...qF...
0000009F 38 46 4b 7c 4e 98 f8 60 03 f1 3f 00 14 ef f0 65 8FK|N..` ..?....e
000000AF a1 7f e7 9f cb c1 4d 0f b8 06 e5 2f 00 85 98 7c .....M. .../...|
000000BF 4c 00 28 00 39 00 38 00 35 00 16 00 13 00 0a 00 L.(.9.8. 5.....
000000CF 33 00 32 00 2f 00 9a 00 99 00 96 00 15 00 12 00 3.2./... .....
000000DF 09 00 14 00 11 00 08 00 06 00 ff 01 00 00 04 00 .....
000000EF 23 00 00 .....#..
00000030 16 fe ff 00 00 00 00 00 00 00 01 00 3d 02 00 00 .....=...
00000040 31 00 01 00 00 00 00 00 31 fe ff 53 b9 a1 1a a3 1..... 1..S....
00000050 6f 9d 49 e3 b5 7d cf 91 06 37 37 10 4b 79 15 80 o.I..}.. .77.Ky..
```


Pro Packet Trace Editing



Pro Packet Trace Editing

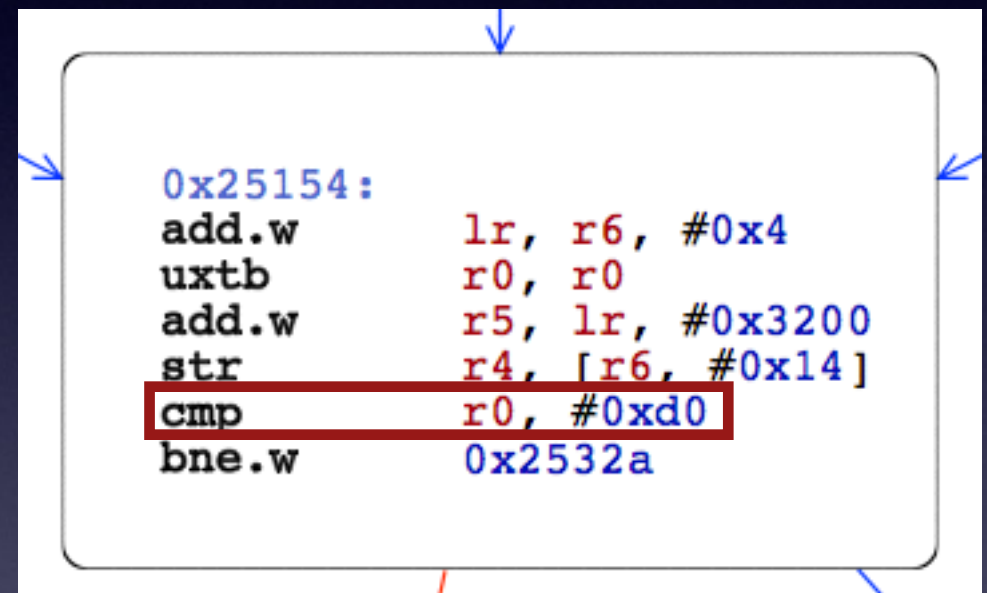
- Success!

Destination	Protocol	Length	Info
MacBook-Pr...	DTLSv1.0	189	Client Hello
192.168.1....	DTLSv1.0	137	Server Hello
192.168.1....	DTLSv1.0	855	Certificate
192.168.1....	DTLSv1.0	72	Certificate Request
192.168.1....	DTLSv1.0	67	Server Hello Done
MacBook-Pr...	DTLSv1.0	855	Certificate
MacBook-Pr...	DTLSv1.0	197	Client Key Exchange
MacBook-Pr...	DTLSv1.0	197	Certificate Verify
MacBook-Pr...	DTLS	60	Continuation Data
MacBook-Pr...	DTLSv1.0	135	Encrypted Handshake Message
192.168.1....	DTLS	57	Continuation Data

Mystery Protocol #2

- DTLS 1.0 with the byte 0xd0 appended to **every** DTLS record

- _gckSessionRecvMessage()



- Inside the DTLS stream
 - Simple plaintext protocol
 - The other peer's PeerID + App data/messages

Bonjour

Advertise local MC service, discover nearby devices advertising the MC service

GCK1 over TCP

Exchange peer names, security options and "candidate" UDP sockets

STUN / ICE

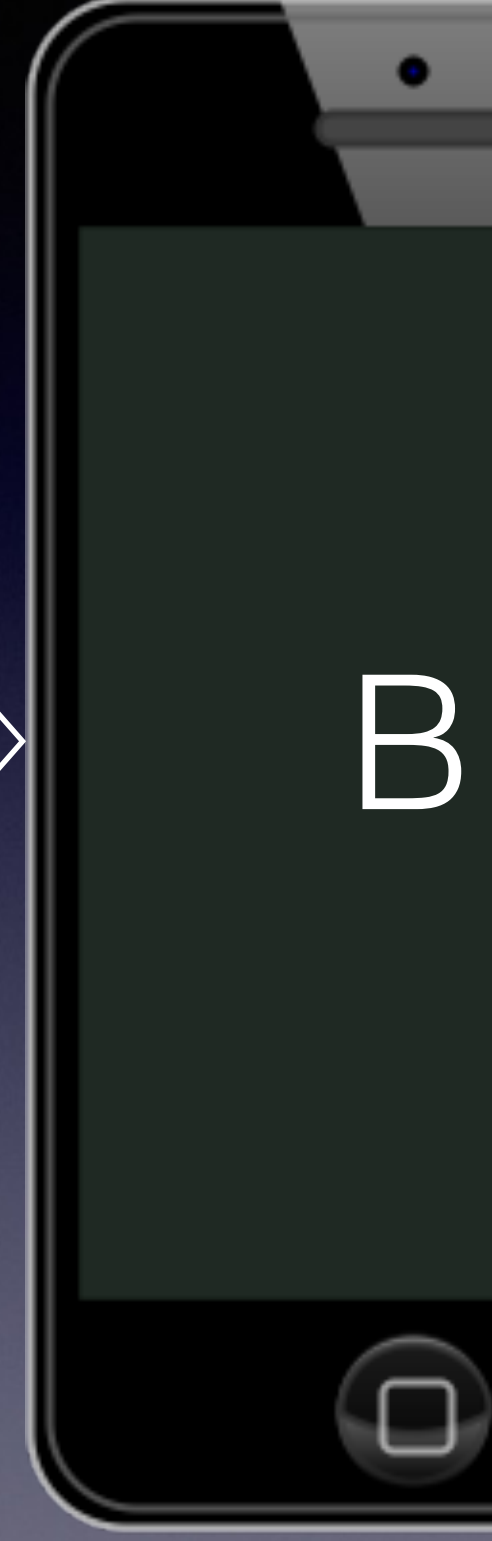
Perform connectivity checks and find the best network path to the other peer

GCK2 over UDP

Perform DTLS handshake, check the other peer's identity, exchange data

A

B



Bonjour

Advertise local MC service, discover nearby devices advertising the MC service

GCK1 over TCP Discovery Phase

Exchange peer names, security options and network information

STUN / ICE

Perform connectivity checks and find the best network path to the other peer

GCK2 over UDP Session Phase

Perform DTLS handshake, check the other peer's identity, exchange data

Security Analysis of Multipeer Connectivity

MC Security Analysis

	MCEncryption None	MCEncryption Optional	MCEncryption Required
Without Authentication			
With Authentication			

MC Security Analysis

	MCEncryption None	MCEncryption Optional	MCEncryption Required
Without Authentication			
With Authentication			

MC Security Analysis

- **MCEncryptionRequired With Authentication:**
DTLS with **mutual** authentication
 - Each peer sends their certificate and validate the other side's certificate
 - RSA & EC-DSA TLS Cipher Suites
 - 30 cipher suites supported in total including PFS cipher suites.
 - In practice, **TLS_RSA_WITH_AES_256_CBC_SHA256** is always negotiated, which doesn't provide PFS

MC Security Analysis

	MCEncryption None	MCEncryption Optional	MCEncryption Required
Without Authentication			
With Authentication			No PFS

MC Security Analysis

	MCEncryption None	MCEncryption Optional	MCEncryption Required
Without Authentication			
With Authentication			No PFS

MC Security Analysis

- **MCEncryptionRequired Without Authentication:**
DTLS with Anonymous TLS Cipher Suites
 - No certificates exchanged
 - “Anon” AES TLS cipher suites:
 - TLS_DH_anon_WITH_AES_128_CBC_SHA,
TLS_DH_anon_WITH_AES_256_CBC_SHA,
TLS_DH_anon_WITH_AES_128_CBC_SHA256,
TLS_DH_anon_WITH_AES_256_CBC_SHA256

MC Security Analysis

	MCEncryption None	MCEncryption Optional	MCEncryption Required
Without Authentication			MiTM
With Authentication			No PFS

MC Security Analysis

	MCEncryption None	MCEncryption Optional	MCEncryption Required
Without Authentication			MiTM
With Authentication			No PFS

MC Security Analysis

- **MCEncryptionNone Without Authentication:**
No DTLS - Plaintext GCK2 protocol

MC Security Analysis

	MCEncryption None	MCEncryption Optional	MCEncryption Required
Without Authentication	Plaintext		MiTM
With Authentication			No PFS

MC Security Analysis

	MCEncryption None	MCEncryption Optional	MCEncryption Required
Without Authentication	Plaintext		MiTM
With Authentication			No PFS

MC Security Analysis

- **MCEncryptionNone With Authentication:**
DTLS with **mutual** authentication
 - Each peer send their certificate and validate the other side's certificate
 - Plaintext / “No Encryption” TLS Cipher Suites!
 - TLS_RSA_WITH_NULL_SHA ,
TLS_RSA_WITH_NULL_SHA256

MC Security Analysis

	MCEncryption None	MCEncryption Optional	MCEncryption Required
Without Authentication	Plaintext		MiTM
With Authentication	Plaintext		No PFS

MC Security Analysis

	MCEncryption None	MCEncryption Optional	MCEncryption Required
Without Authentication	Plaintext		MiTM
With Authentication	Plaintext		No PFS

MC Security Analysis

- **MCEncryptionOptional With Authentication**
- "The session **prefers** to use encryption, but will accept unencrypted connections"

Conclusion

	MCEncryption None	MCEncryption Optional	MCEncryption Required
Without Authentication	Plaintext	MitM	MitM
With Authentication	Plaintext		No PFS

Conclusion

	MCEncryption None	MCEncryption Optional	MCEncryption Required
Without Authentication	Plaintext	MitM	MitM
With Authentication	Plaintext		No PFS

MC Security Analysis

- **MCEncryptionOptional With Authentication**
- "The session **prefers** to use encryption, but will accept unencrypted connections"
- Two peers using MCEncryptionOptional with Authentication should get the same security as MCEncryptionRequired (ie. use DTLS)
- Authentication should prevent a man-in-the-middle from tampering with the network traffic

Bonjour

Advertise local MC service, discover nearby devices advertising the MC service

GCK1 over TCP

Exchange peer names, security options and "candidate" UDP sockets

STUN / ICE

Perform connectivity checks and find the best network path to the other peer

GCK2 over UDP

Perform DTLS handshake, check the other peer's identity, exchange data



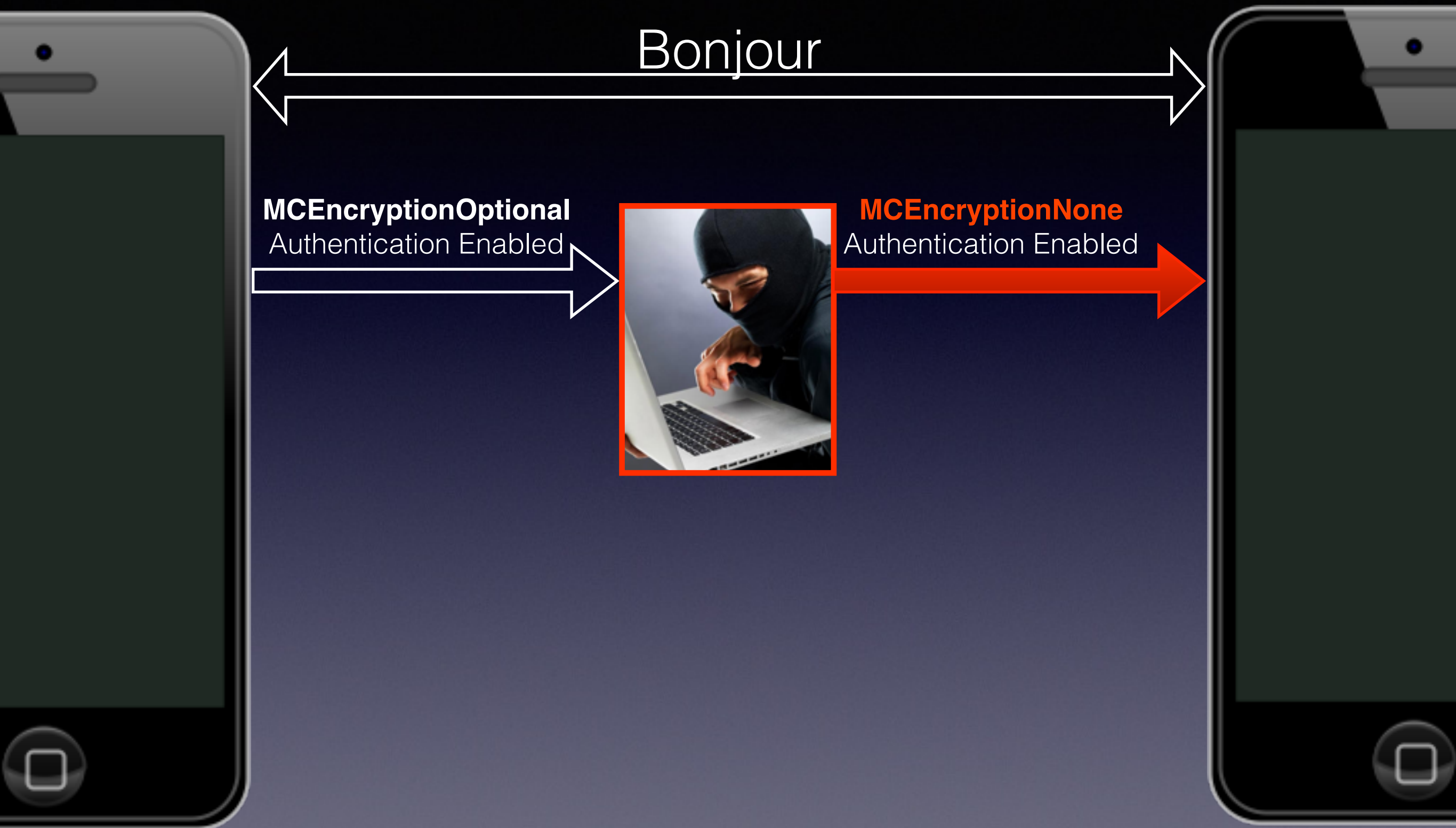
Bonjour

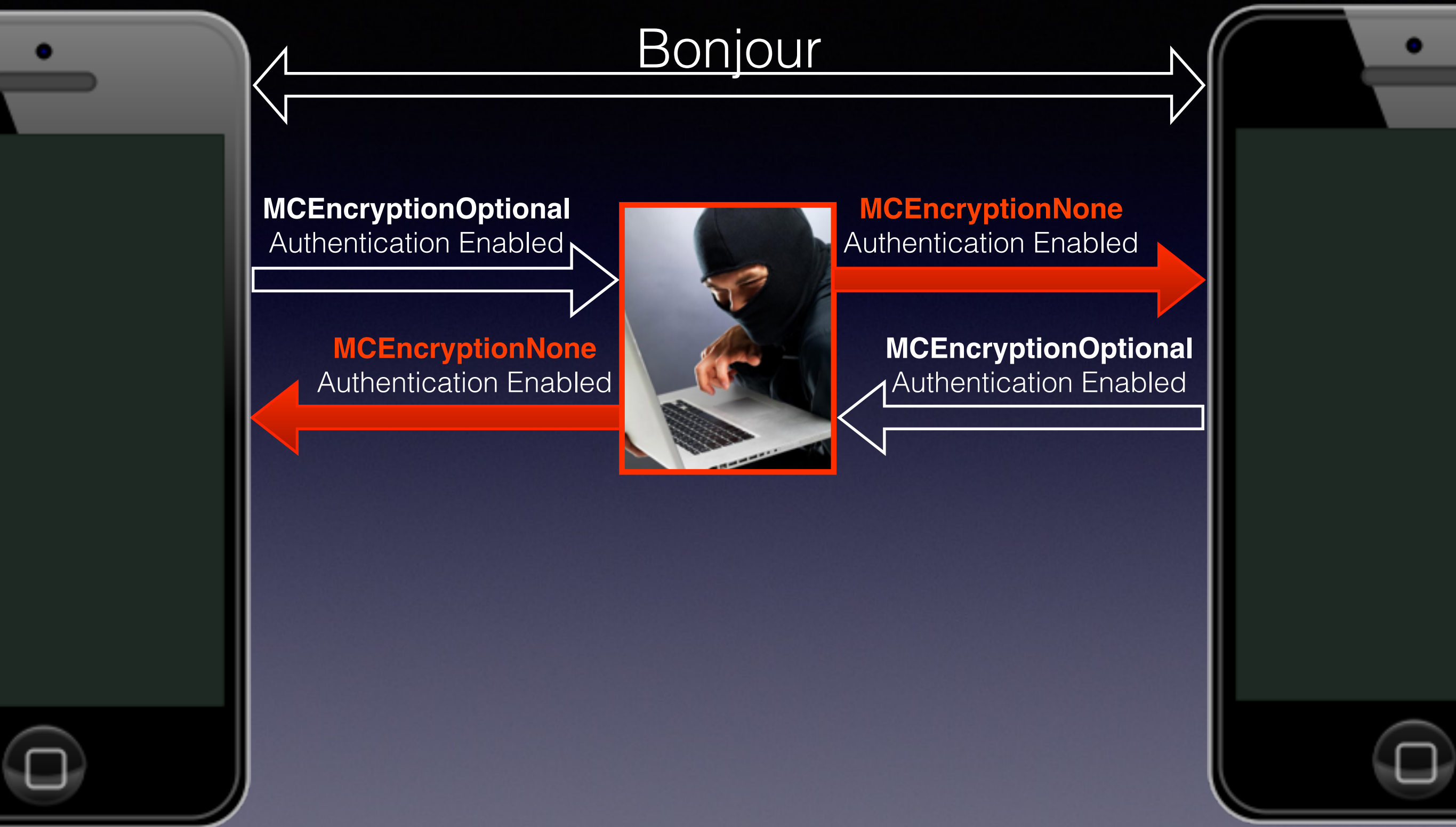


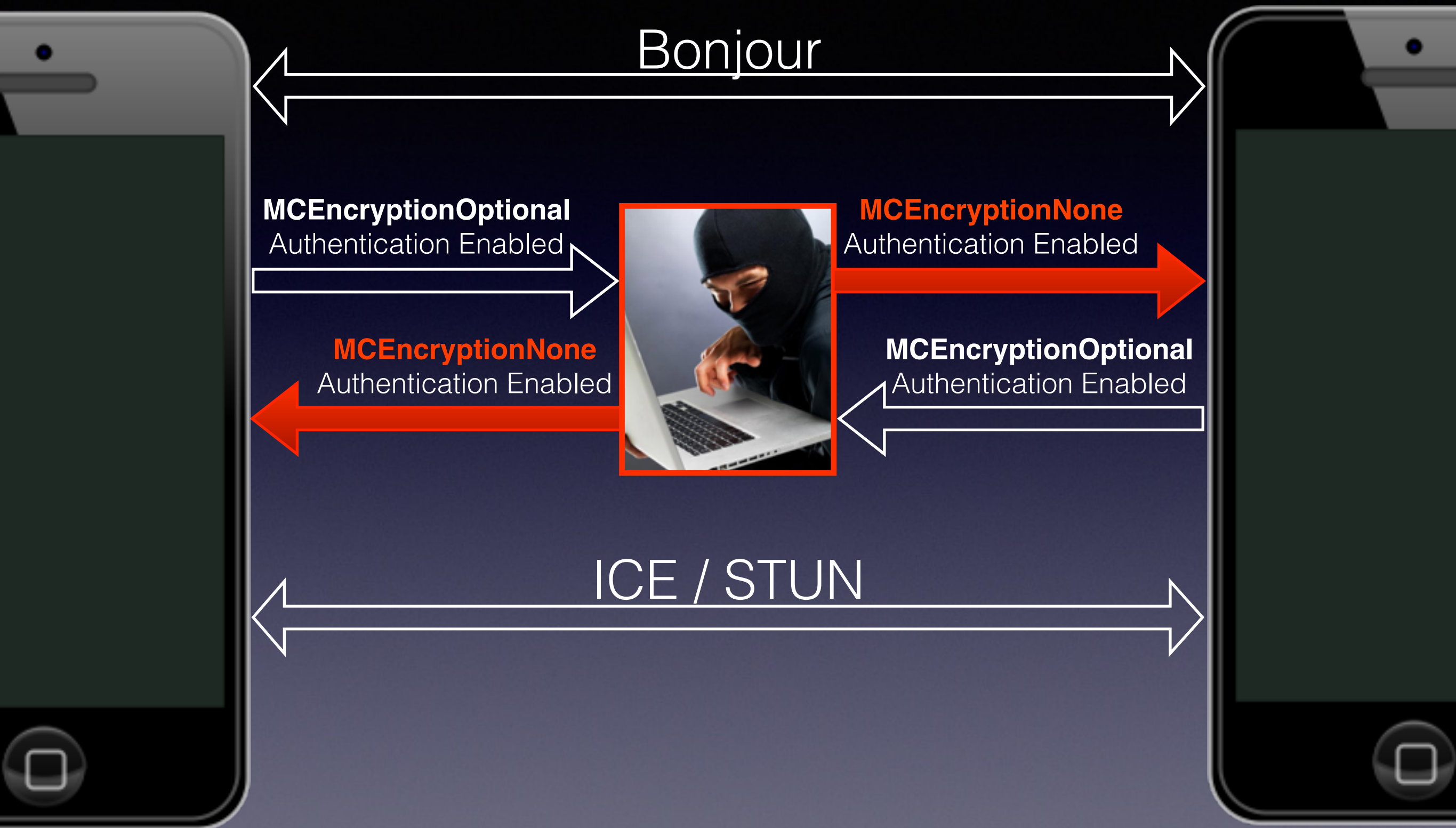
Bonjour

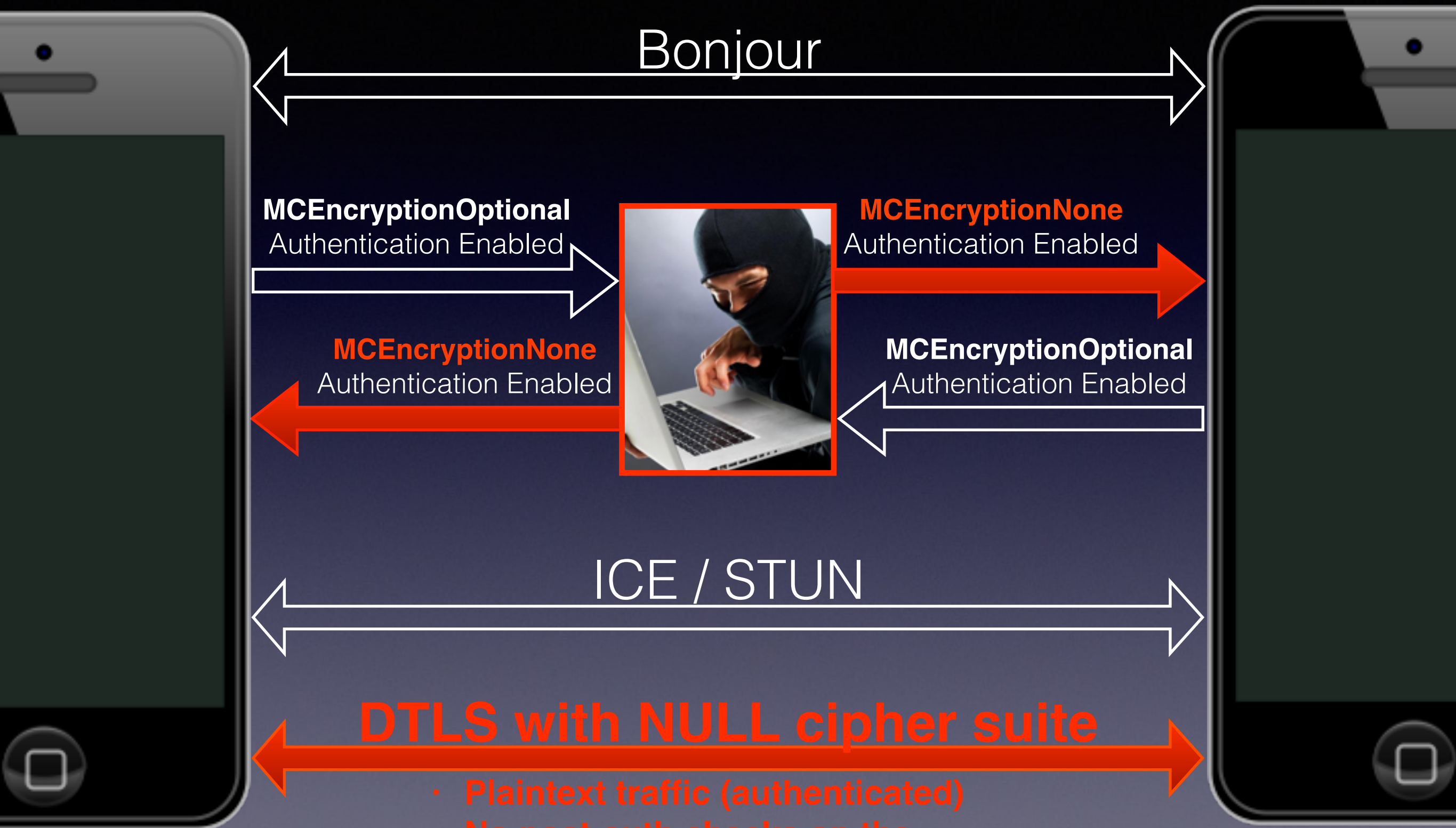
MCEncryptionOptional
Authentication Enabled











- Plaintext traffic (authenticated)
- No post-auth checks on the MCEncryption parameters exchanged
- Same security as MCEncryptionNone

MCEncryptionOptional Downgrade Attack

00000A4F	d0 17 fe ff	00 01 00 00	00 00 00 19	00 46 c1 05	F..
00000A5F	00 32 00 00	cf f0 7e 0c	0c 44 6f bb	0f 19 01 02	.2....~. .Do.....
00000A6F	43 2e 74 68	69 73 20 69	73 20 61 20	74 65 73 74	C.this i s a test
00000A7F	20 6d 65 73	73 61 67 65	20 31 32 33	20 31 32 33	message 123 123
00000A8F	80 d1 dd f4	13 89 45 80	36 ad 2b 4f	f4 0e dc e4	E. 6.+0....
00000A9F	b4 2e 2f 32				../2
00000B8F	d0 17 fe ff	00 01 00 00	00 00 00 1a	00 28 c1 08	(..
00000B9F	00 14 00 00	1c 2b 6f bb	0f 19 7e 0c	0c 44 00 02	+o. ..~..D..
00000BAF	00 01 e0 aa	33 77 44 53	91 6f 63 a6	85 4f f5 50	3wDS .oc..0.P
00000BBF	f8 73 48 2e	bc 35			.sH..5

MC Security Analysis

	MCEncryption None	MCEncryption Optional	MCEncryption Required
Without Authentication	Plaintext	MitM	MitM
With Authentication	Plaintext	MitM (Downgrade)	No PFS

Conclusion

Conclusion

- Most security settings work as advertised by the MC API
 - **Except for** MCEncryptionOptional with Authentication
- Some combinations should never be used
 - MCEncryptionOptional
 - MCEncryptionNone with Authentication
- Only MCEncryptionRequired with Authentication is secure

Conclusion

	MCEncryption None	MCEncryption Optional	MCEncryption Required
Without Authentication	Plaintext	MitM	MitM
With Authentication	Plaintext	MitM (Downgrade)	No PFS

Conclusion

	MCEncryption None	MCEncryption Optional	MCEncryption Required
Without Authentication	Plaintext	MitM	MitM
With Authentication	Plaintext	MitM (Downgrade)	No PFS

Conclusion

- Possible improvements to the MC Framework:
 - MCEncryptionRequired with Authentication:
 - Prioritize PFS TLS Cipher Suites
 - MCEncryptionOptional with Authentication:
 - Peers should validate security parameters post-authentication to prevent downgrade attacks
 - Better: remove MCEncryptionOptional and make MCEncryptionRequired the default setting?

Thanks!

More at

<https://nabla-c0d3.github.io>