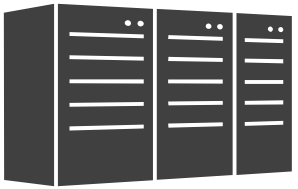
Part IV
Forward Secrecy

8th BIU Winter School on Key Exchange, 2018

Marc Fischlin

What is wrong with the following Protocol?

$N_S, \mathrm{pk}_S, \mathrm{cert}_S$



pick random K

$C \leftarrow \mathrm{Enc}(\mathrm{pk}_S, N_S|K)$

C

$N_S|K \leftarrow \mathrm{Dec}(\mathrm{sk}_S, C)$

check N_S

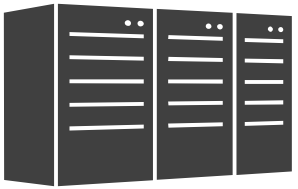
It is BR-secure against active attacks!

(Why?)

Note: We only claim unilateral authentication

It is not Forward-Secret!

$N_S, \mathrm{pk}_S, \mathrm{cert}_S$



pick random K
$C \leftarrow \mathrm{Enc}(\mathrm{pk}_S, N_S|K)$

C

$N_S|K \leftarrow \mathrm{Dec}(\mathrm{sk}_S, C)$
check N_S

If adversary later breaks encryption scheme (corrupt query!) then it can recover session key.

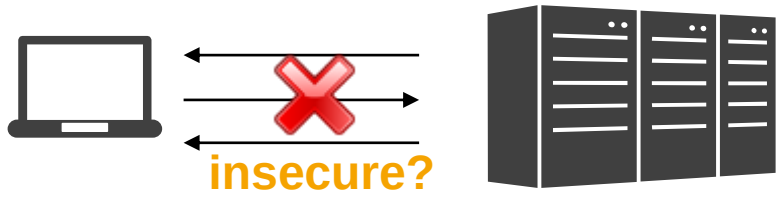
So far, in our model the sessions would not be fresh anymore.

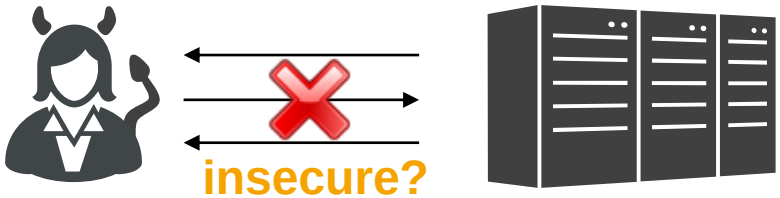
Forward-Secrecy in BR-Model

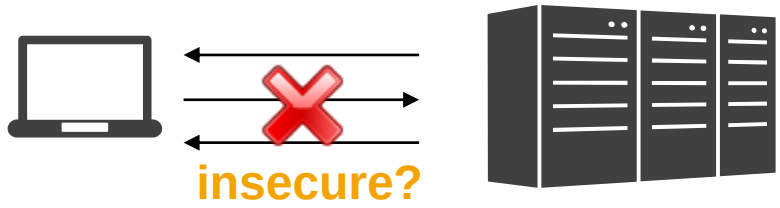
Mutual Authentication	Unilateral Authentication	Anonymous
neither TEST session nor partner session REVEALED	…	…
neither party in TEST nor intended partner pid CORRUPT **before session complete**	+	+
	if unauthenticated partner then there is honest partner session **(partner may be corrupted later)**	there is honest partner session **(partner may be corrupted later)**

Interpretation

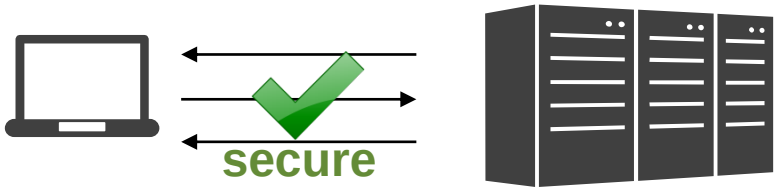
non-forward secret

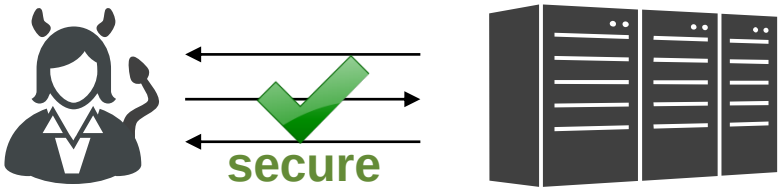






forward secret



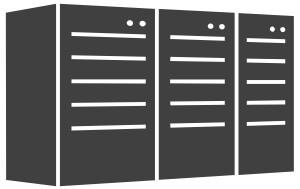


Recovering single long-term secret enough

Gets long-term secret later

Would need to recover many ephemeral secrets

TLS 1.3: (EC)DHE-Handshake and FS



CH
CKS

$r_c \leftarrow \{0,1\}^{256}$

g^x

$r_s \leftarrow \{0,1\}^{256}$

g^y

SH
SKS

handshake key
$\leftarrow \mathrm{KDF}(g^{xy}, \mathtt{CH} \ldots \mathtt{SKS})$

handshake key
$\leftarrow \mathrm{KDF}(g^{xy}, \mathtt{CH} \ldots \mathtt{SKS})$

{SConf*}
pk_S, cert_S {SCert*}
s {SCertV*}
t {SF}

$s \leftarrow \mathrm{Sign}(\mathrm{sk}_s, \mathtt{CH} \ldots \mathtt{SCert})$

$t \leftarrow \mathrm{MAC}(\mathrm{k}_{\mathrm{SF}}, \mathtt{CH} \ldots \mathtt{SCert})$

channel key is FS, because it is derived from ephemeral DH, and signature key cannot be corrupt when key is derived

channel key
$\leftarrow \mathrm{KDF}(g^{xy}, \mathtt{CH} \ldots \mathtt{C}$

channel key
$\leftarrow \mathrm{KDF}(g^{xy}, \mathtt{CH} \ldots \mathtt{CF})$

Two-Move Protocols and (weak) Forward Secrecy

Forward Secrecy and Two-Move Protocols

"MQV does not provide Perfect Forward Secrecy (PFS). This, however, is not just a failure of MQV but it's an inherent limitation of implicitly-authenticated 2-message protocols based on public-key authentication (and which do not rely on a previously established shared state between the parties).
Indeed no such protocol can provide PFS."

Krawczyk: HMQV: A High-Performance Secure Diffie-Hellman Protocol, eprint, 2005

Several prerequisites in impossibility result:

…implicitly-authenticated… *…public-key authentication…* *…no (shared) state…*

According to which security model?

Attacking Two-Move Protocols

implicitly authenticated

long-term secret x_C

randomness r_C

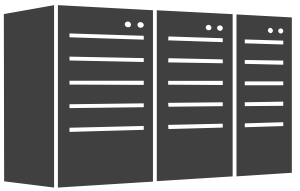
K from x_C, r_C, m_S

implicitly authenticated:
x_C not used here
(e.g., in signature)

m_C →

← m_S

long-term secret x_S



randomness r_S

K from x_S, r_S, m_C

randomness r_C^*

m_C^* →

← m_S'

randomness r_S'

K' from x_S, r_S', m_C^*

1.corrupt $\rightarrow$ x_C

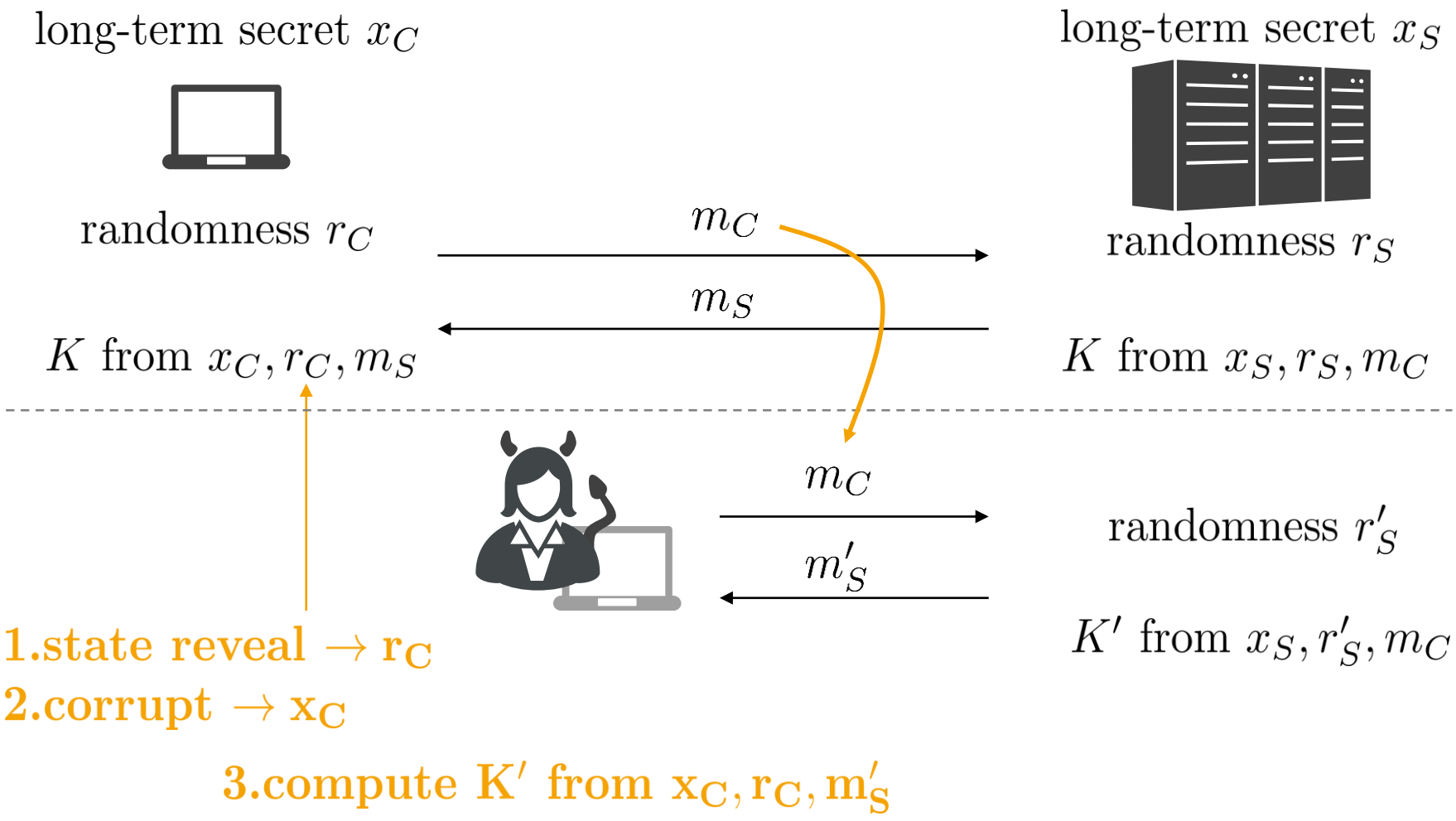
2.compute K' from x_C, r_C^*, m_S'

Bellare, Pointcheval, Rogaway: Authenticated Key Exchange Secure Against Dictionary Attacks, Eurocrypt 2000

Attacking General Two-Move Protocols

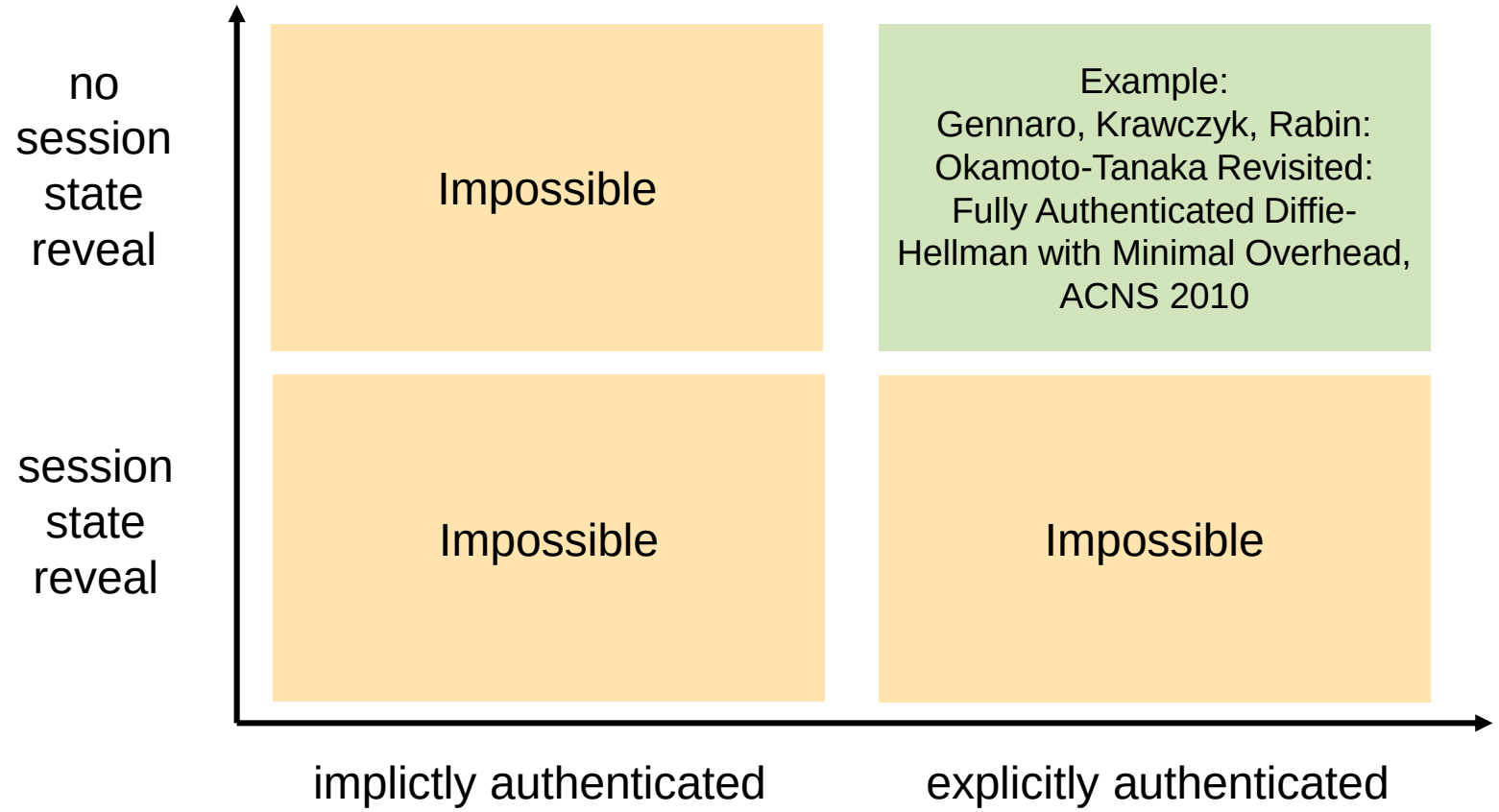
w/ state reveal

Boyd, Gonzales Nieto: On forward secrecy in one-round key exchange, Cryptography and Coding, 2011



On the Possibility of FS for Two-Move Schemes

Public-key based and *stateless* schemes:



Bellare, Pointcheval, Rogaway: Authenticated Key Exchange Secure Against Dictionary Attacks, Eurocrypt 2000

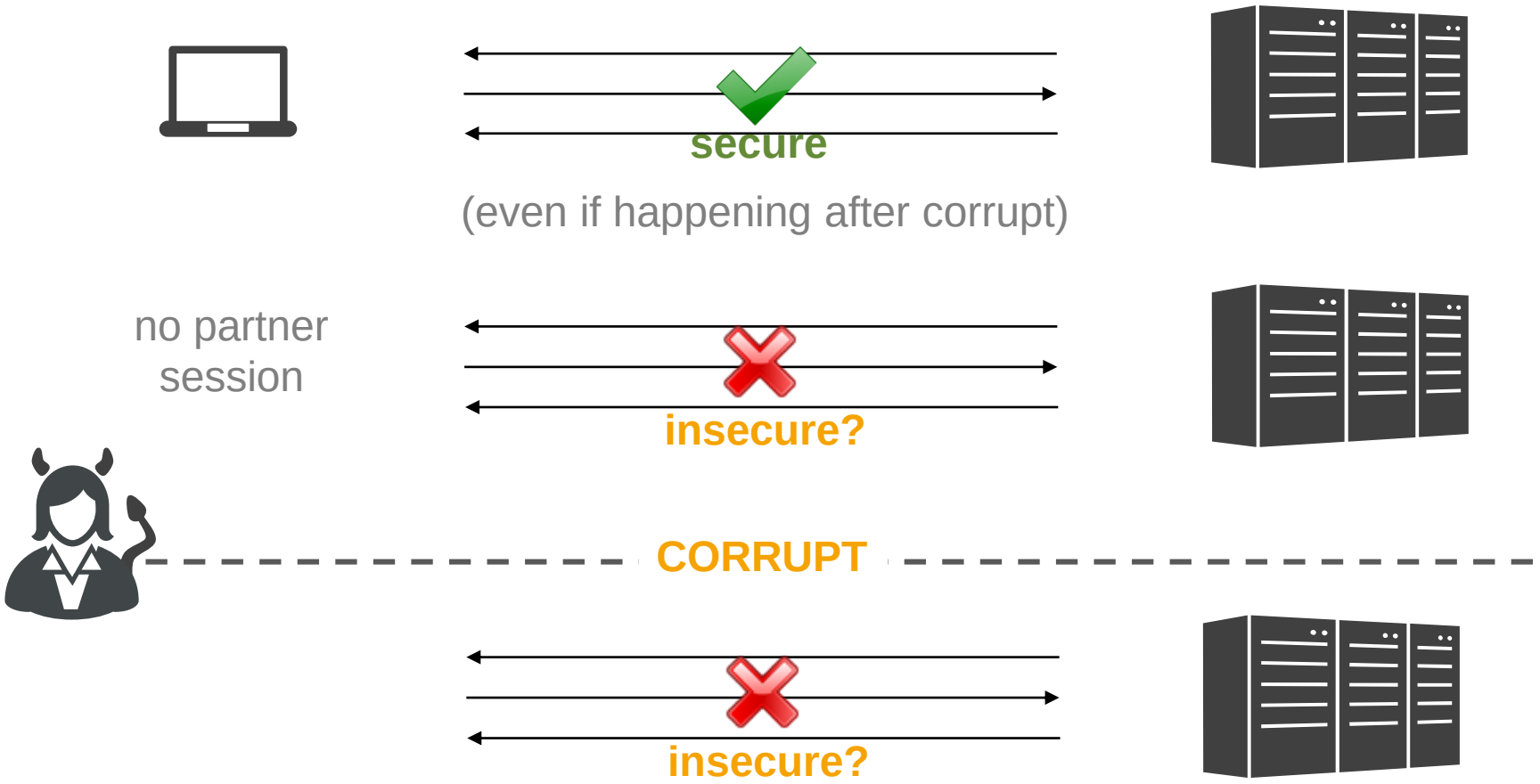
Relaxation to Weak Forward Secrecy

Krawczyk: HMQV: A High-Performance Secure Diffie-Hellman Protocol, Crypto 2005

weak [Perfect] Forward Secrecy (wFS):

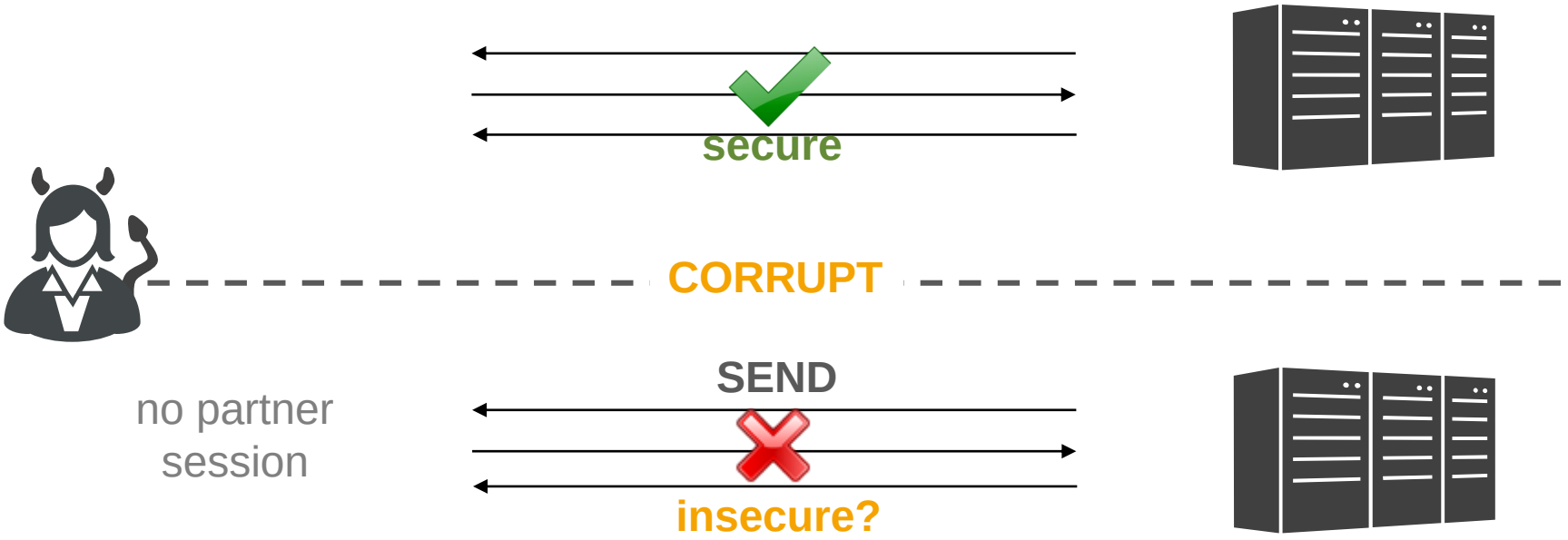
Sessions in which
adversary did not interfere with execution
are still considered fresh
(before *and* after corruption)

Weak [Perfect] Forward Secrecy



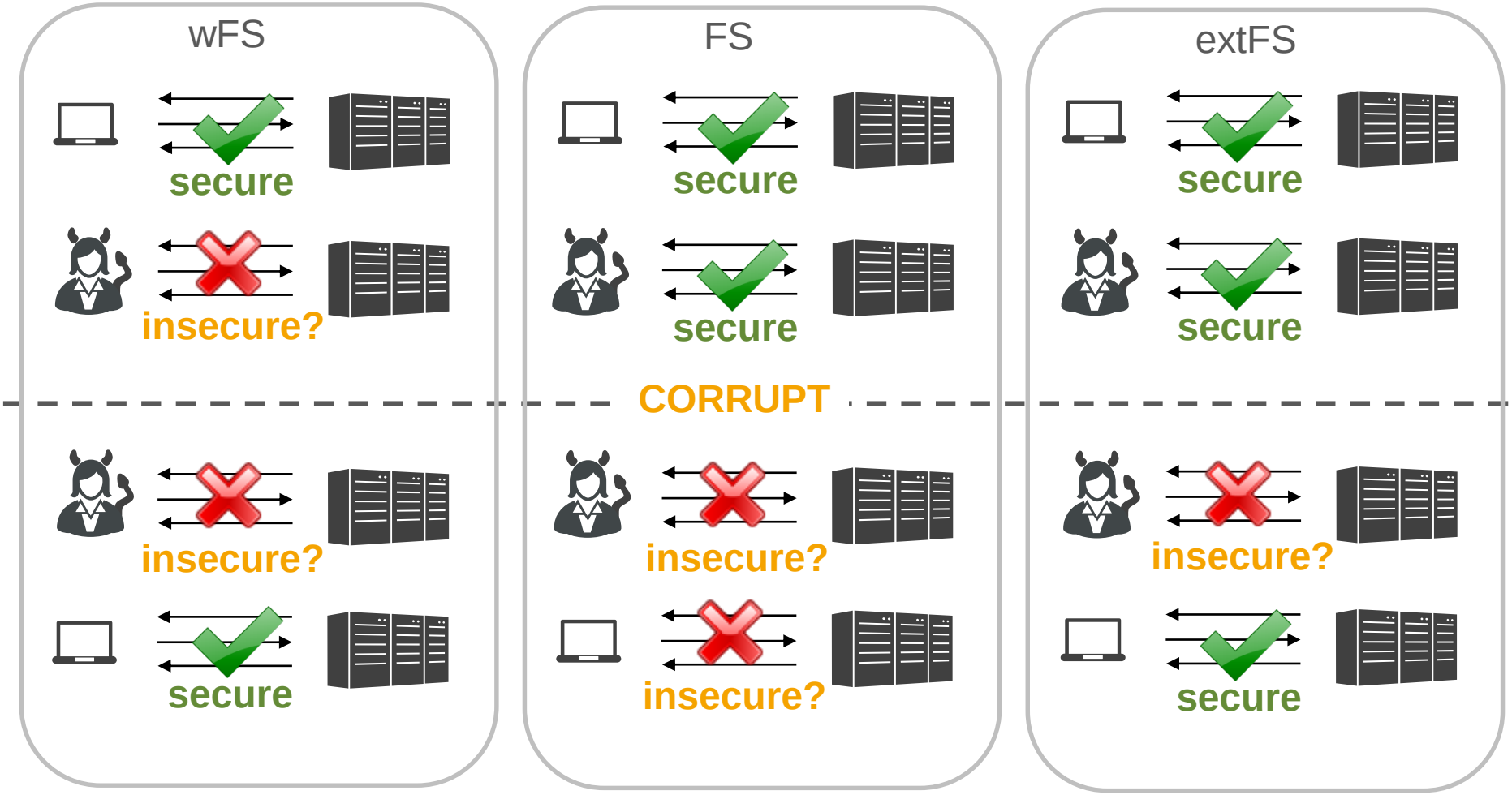
Comparison of FS Notions

FS-“Extension“ in BPR00-Model

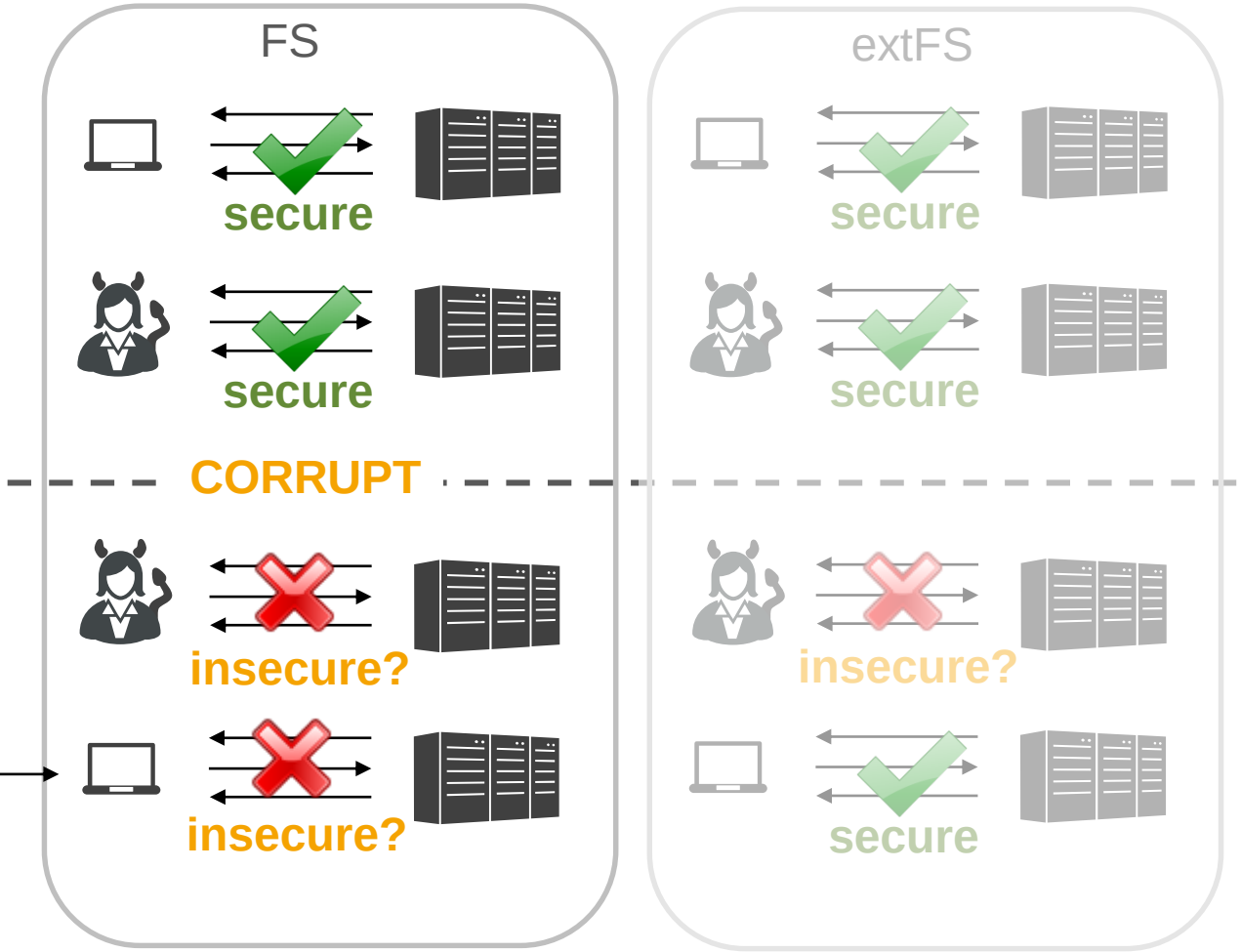


Freshness according to BPR00 also demands that future „honest sessions“ are secure:

Comparison of FS Notions



Teaser for the Break



Think of a protocol for which honest execution via EXEC after corrupt is insecure