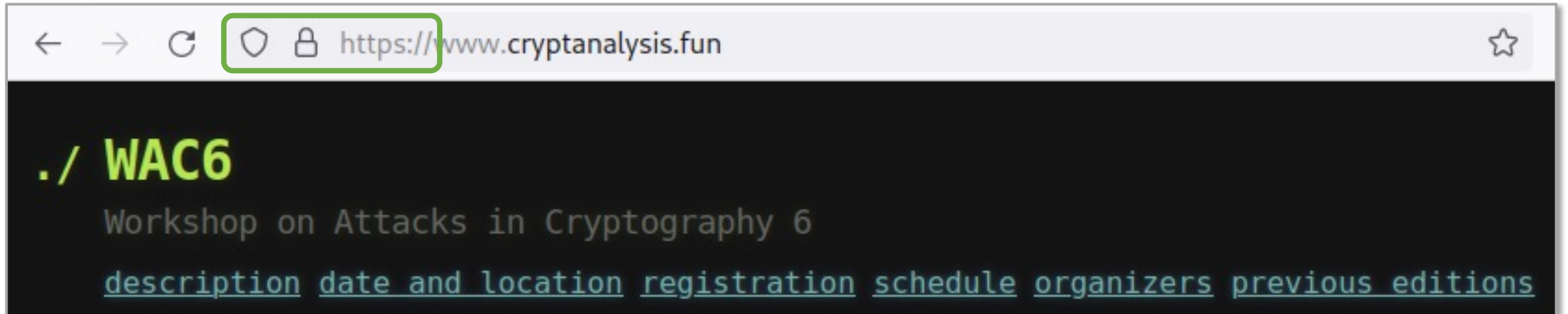


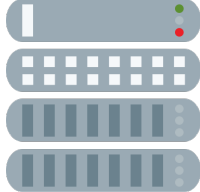
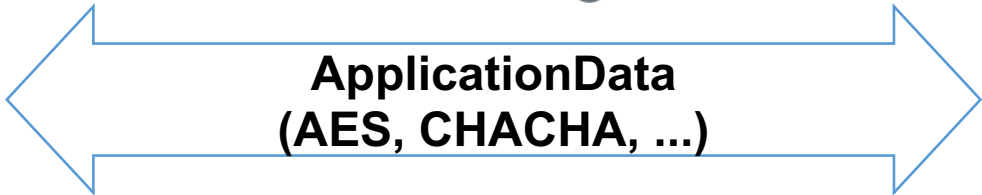
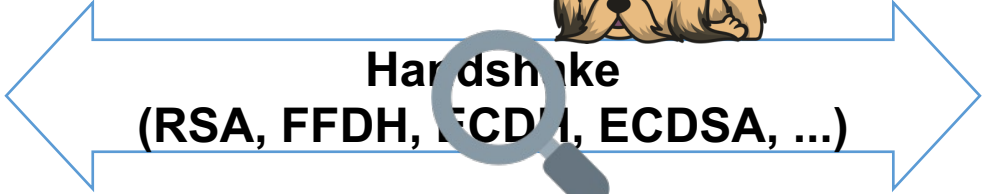
We Really Need to Talk About Session Tickets

Sven Hebrok, Simon Nachtigall, Marcel Maehren, Nurullah Erinola,
Robert Merget, Juraj Somorovsky, and Jörg Schwenk

TLS is Widely Used



TLS Handshake

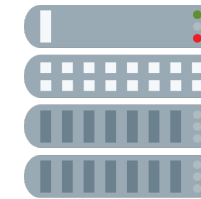


Master Secret

Session Keys



TLS Handshake Internally



Hello

Hello

Certificate 

Key Exchange 

Hello Done

Key Exchange 

...

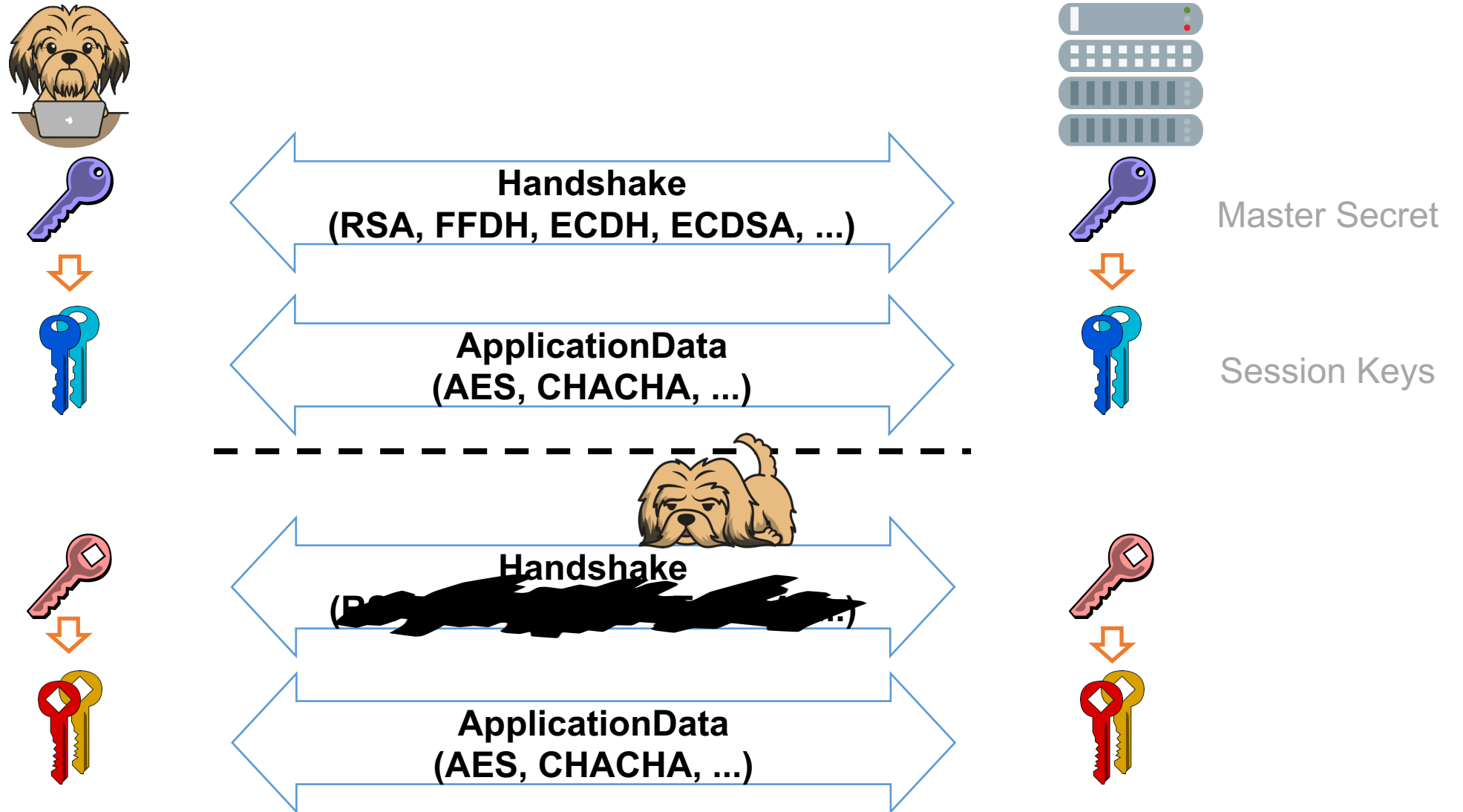
Application



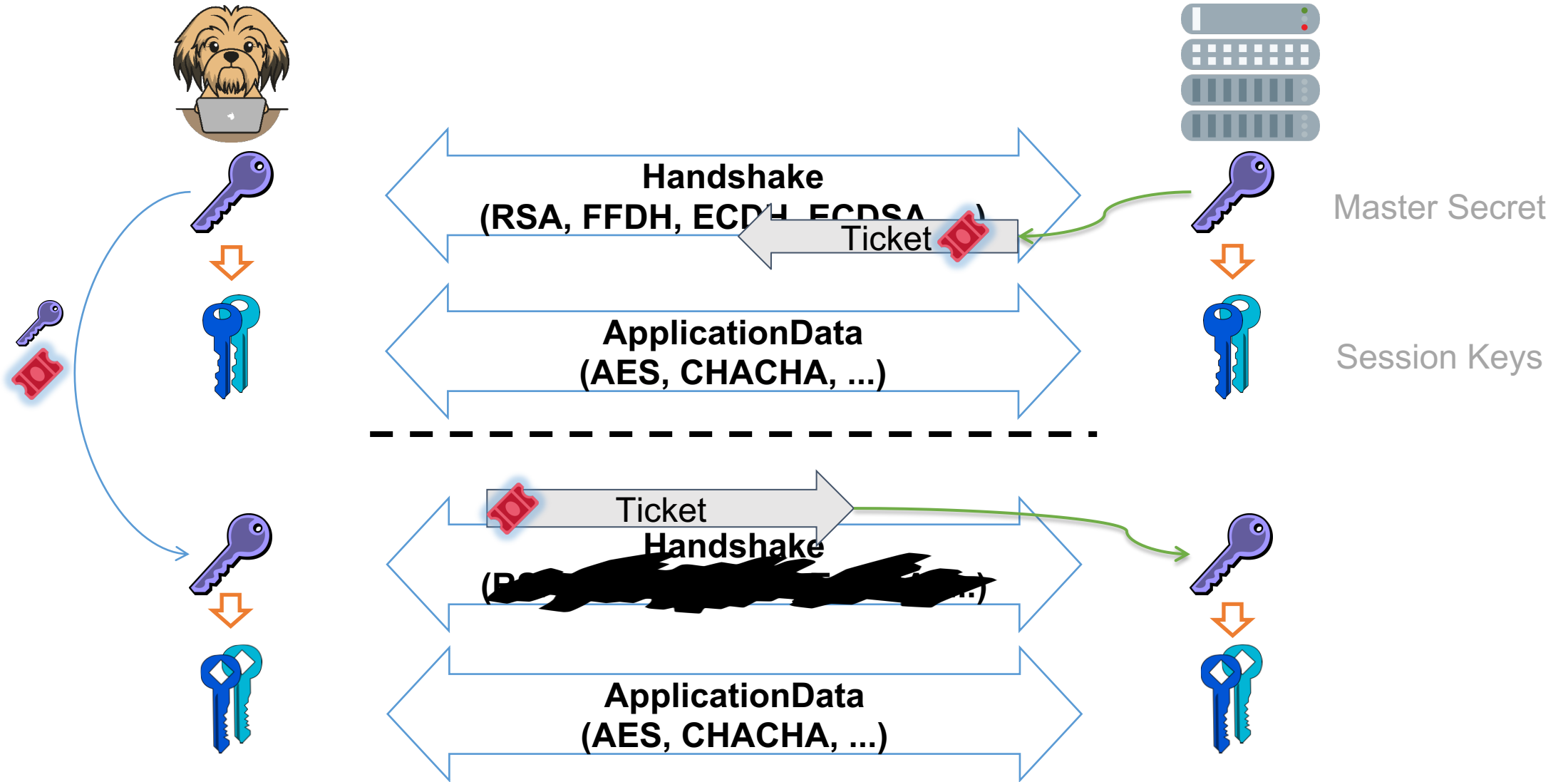
(EC)DHE
(EC)DH
RSA

RSA/DSS
Signature

TLS Session Resumption

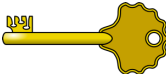




TLS Session Resumption using Tickets



What is a Ticket?

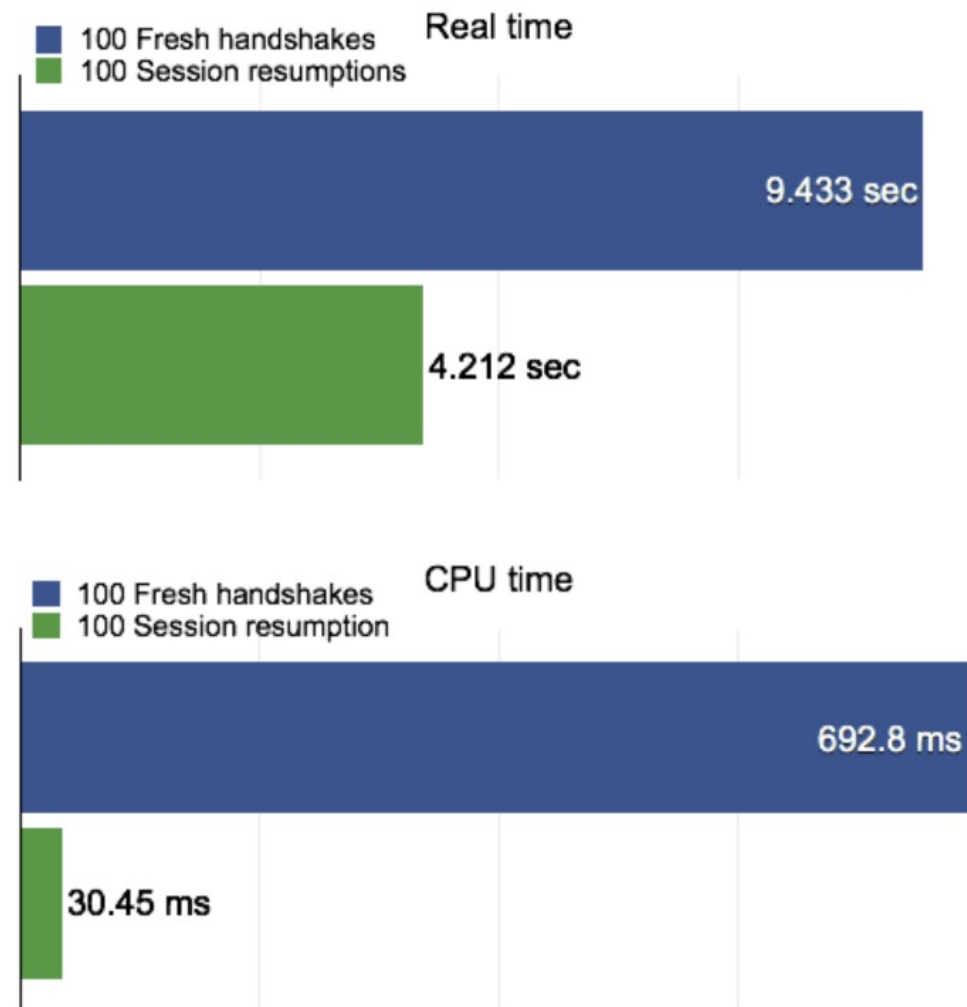
$$\text{Ticket} = \text{Enc}_{\text{STEK}}(\text{Key})$$

Server stores one  - used for all clients
Client stores  and 

 STEK (Session Ticket Encryption Key)

Session Tickets Boost Performance

- near 100% Browser Support
- ~75% Server Support
- -50% connection establishment time
- -95% CPU time



Session Tickets Have Known Issues

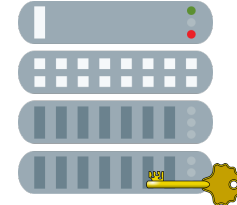
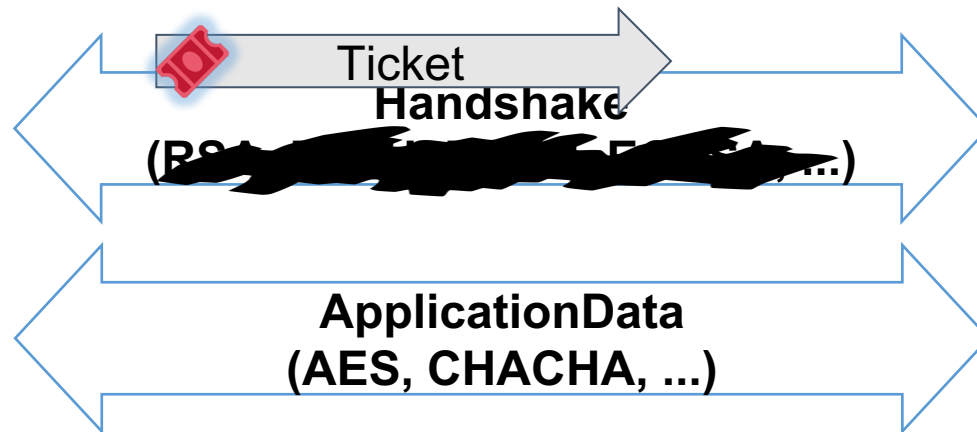
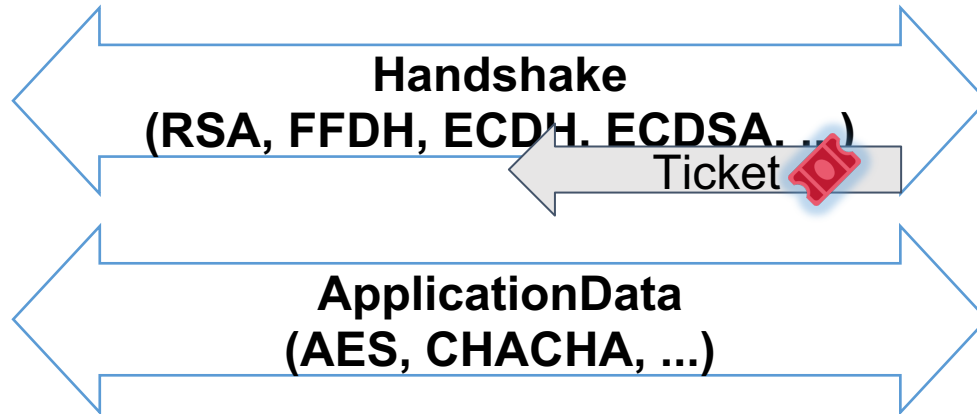


28 Sep 2017

**WE NEED TO TALK ABOUT SESSION
TICKETS**

<https://words.filippo.io/we-need-to-talk-about-session-tickets/>

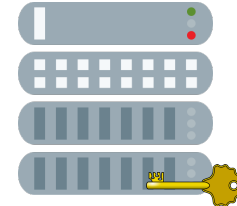
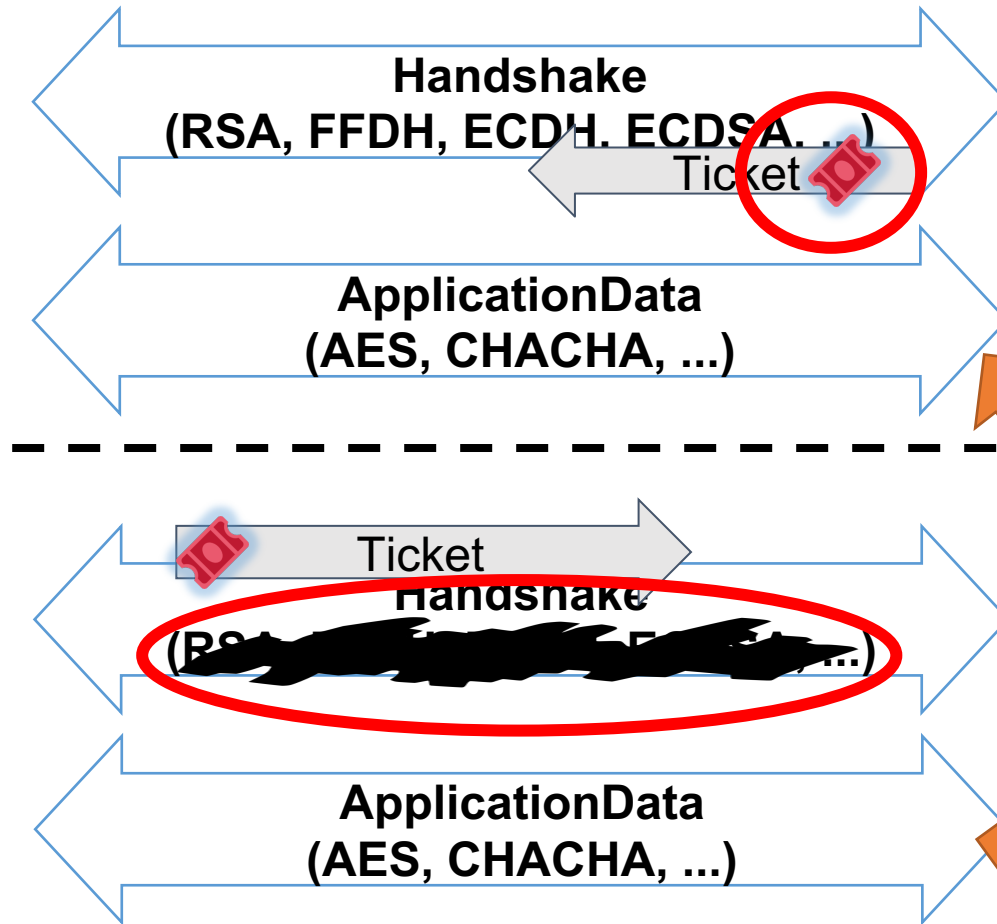
Session Tickets Have Known Issues



Issues & Impact of STEK compromise

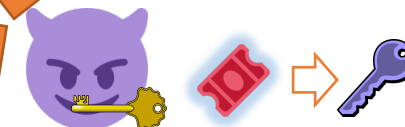


Session Tickets Have Known Issues



Issues & Impact of STEK compromise

1. No key exchange
⇒ Can decrypt passively
⇒ Can impersonate actively
2. Same secret reused
⇒ Can decrypt previous sessions
3. Tickets sent in plaintext
⇒ Can decrypt first connection immediately



TLS Session Tickets

Brief Summary

- Speed up handshake
 - No key exchange
 - No certificate
- Encrypted using STEK
 - Known only to server
- STEK compromise catastrophic
 - Passive traffic decryption
 - Decrypt previous and future sessions
 - Active server impersonation

We Really Need to Talk About Session Tickets: A Large-Scale Analysis of Cryptographic Dangers with TLS Session Tickets

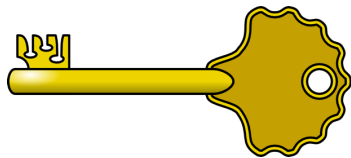
Authors:

Sven Hebrok, *Paderborn University*; Simon Nachtigall, *Paderborn University and achelos GmbH*; Marcel Maehren and Nurullah Erinola, *Ruhr University Bochum*; Robert Merget, *Technology Innovation Institute and Ruhr University Bochum*; Juraj Somorovsky, *Paderborn University*; Jörg Schwenk, *Ruhr University Bochum*



Motivation: GnuTLS

CVE-ID	
CVE-2020-13777	Learn more at National Vulnerability Database (NVD) • CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information
Description	
GnuTLS 3.6.x before 3.6.14 uses incorrect cryptography for encrypting a session ticket (a loss of confidentiality in TLS 1.2, and an authentication bypass in TLS 1.3). The earliest affected version is 3.6.4 (2018-09-24) because of an error in a 2018-09-18 commit. Until the first key rotation, the TLS server always uses wrong data in place of an encryption key derived from an application.	



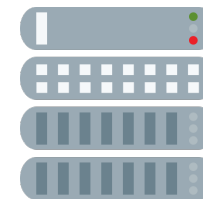
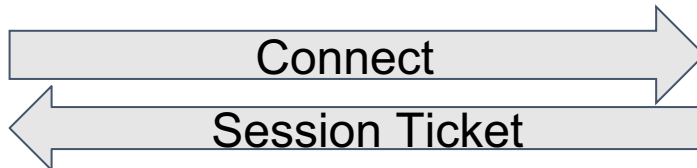
= 0x00

How widespread is something like this?

→ Scan servers in the wild

Live Demo

Ticket Format?



How to determine whether STEK=000000?

- Handshake Protocol: New Session Ticket

Handshake

Length: 23

→ decrypt with key=000000

- TLS Session Ticket

What to decrypt?

800 seconds (1 day, 4 hours)

Where's the IV?

06107535a8f78ad158f3134dca563e7...

- TLSv1

Spec Protocol: Change Cipher Spec

- TLSv1

Where's the Ciphertext?

Protocol: Encrypted Handshake Message

Con

Version: TLS 1.2 (0x0202)

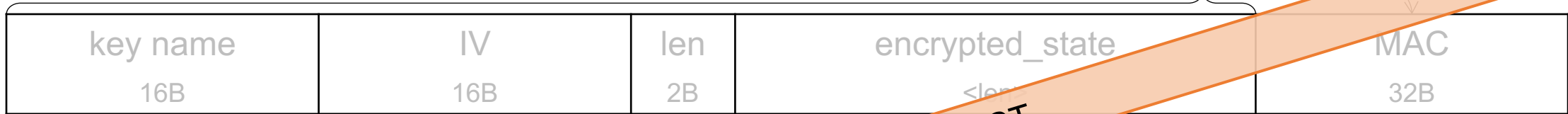
```
0050 c0 00 e5 02 d3 2h 3f 67 3h a5 16 10 61 07 53 5a ... ..+?g ;...a.SZ
0060 8f 78 ad 15 ... ..x...14. .c.p1'ms
0070 13 1c 6a 0c ... ..j....q A.`..K.}
0080 6d ac 0d 59 cb 45 48 c6 3e b0 02 3e 4f e7 f7 45 m..Y.EH. >..>0..E
0090 73 28 eb 73 93 58 7a c0 68 f7 c6 d4 d0 20 66 a7 s(.s.Xz. h... f.
00a0 77 8d 97 2e 0f f4 e6 2a d7 a3 59 db b2 a2 36 61 w.....* ..Y...6a
```

How do tickets actually work?

Session Tickets on the Byte Level

 = Enc  ()

- RFC 5077:

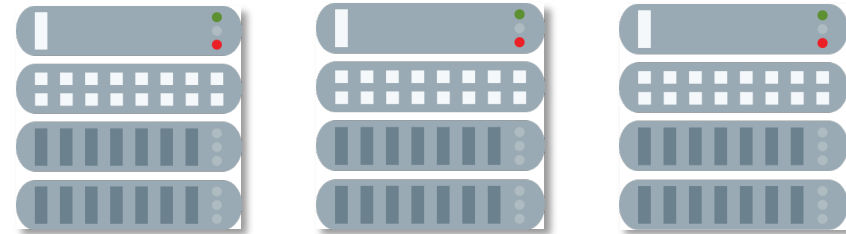


- AES-128-CBC
- HMAC-SHA-256
- Encrypt-then-Mac
- Regular key rotation

only RECOMMENDED, no MUST
Implementations could do anything

Our Plan

```
uint8_t *ptr;
if (!CBB_add_bytes(out, key_name, 16) ||
    !CBB_add_bytes(out, iv, EVP_CIPHER_CTX_iv_length(ctx.get())) ||
    !CBB_reserve(out, &ptr, session_len + EVP_MAX_BLOCK_LENGTH)) {
    return 0;
}
```



1. Analyze open-source implementations

- Ticket format
- Algorithms
- Look for immediate issues

2. Large-scale analysis

Open Source Analysis Algorithms

RFC 5077

AES-128-CBC

HMAC-SHA-256

BoringSSL

AES-128-CBC

HMAC-SHA-256

Open Source Analysis Algorithms

RFC 5077	AES-128-CBC	HMAC-SHA-256
BoringSSL	AES-128-CBC	HMAC-SHA-256
Botan	AES-256-GCM	(GMAC)
GnuTLS	AES-256-CBC	HMAC-SHA-1
GoTLS	AES-128-CTR	HMAC-SHA-256
MatrixSSL (TLS 1.2)	AES-256-CBC	HMAC-SHA-256
MatrixSSL (TLS 1.3)	AES-256-GCM	(GMAC)
mbedTLS	AES-128/256-GCM/CCM	(GMAC/CBCMAC)
OpenSSL	AES-256-CBC	HMAC-SHA-256
Rustls	ChaCha20	Poly1305
s2n	AES-256-GCM	(GMAC)
Apache	AES-128-CBC	HMAC-SHA-256
Nginx	AES-128/256-CBC	HMAC-SHA-256
OpenLiteSpeed	AES-128-CBC	HMAC-SHA-256

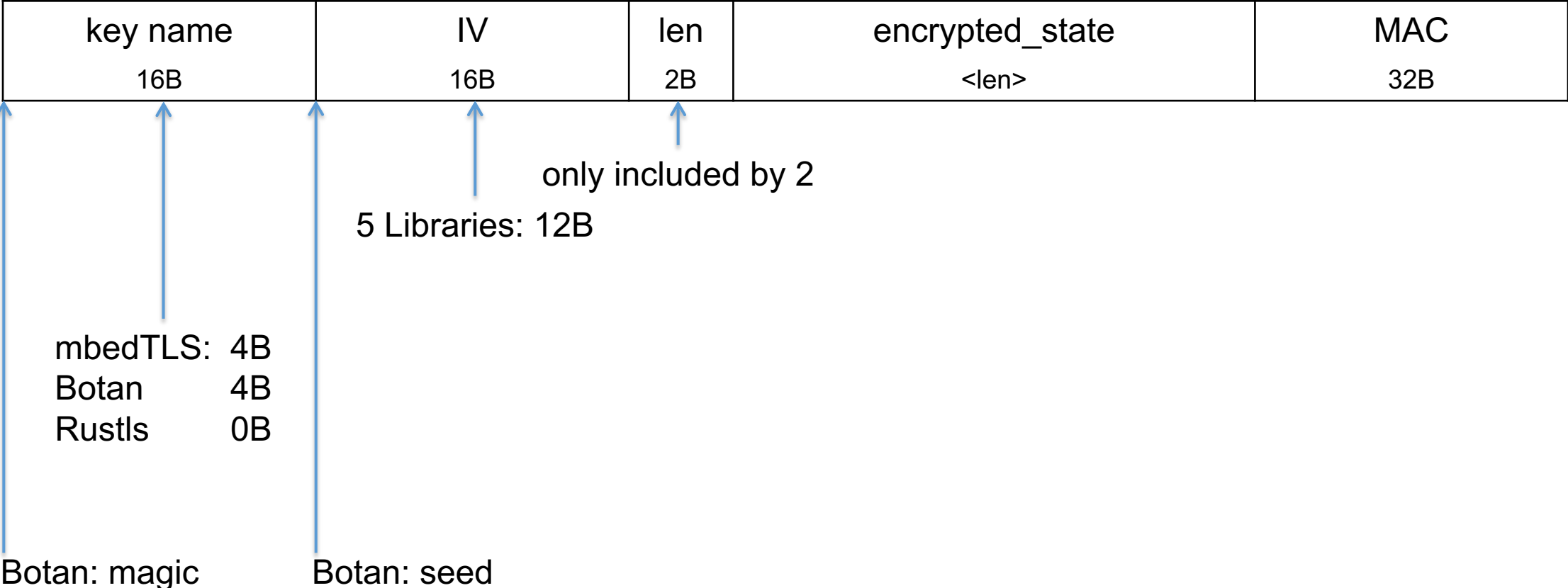
As Recommended

Complexity

Key Wearout Possible - Addressed

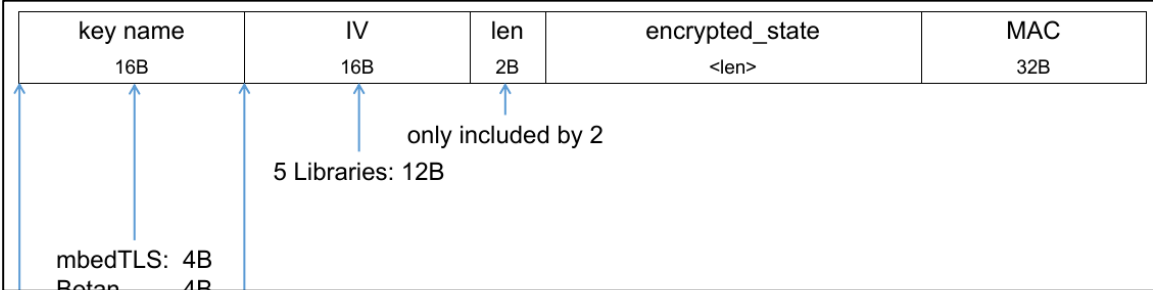
Weaker than Recommended

Open Source Analysis Format



Open Source Analysis Conclusion

RFC 5077	AES-128-CBC	HMAC-SHA-256
BoringSSL	AES-128-CBC	HMAC-SHA-256
Botan	AES-256-GCM	(GMAC)
GnuTLS	AES-256-CBC	HMAC-SHA-1
GoTLS	AES-128-CTR	HMAC-SHA-256
MatrixSSL (TLS 1.2)	AES-256-CBC	HMAC-SHA-256



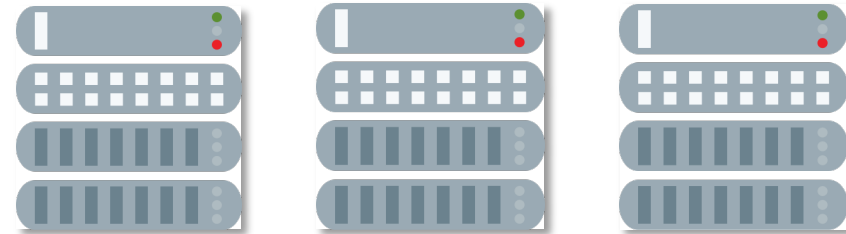
- Some complexity
- One weaker algorithm

- Close to recommended format
 - Some fields shorter
 - IV/Nonce before ciphertext
 - Encrypt-then-MAC

no immediate issues

Our Plan

```
uint8_t *ptr;
if (!CBB_add_bytes(out, key_name, 16) ||
    !CBB_add_bytes(out, iv, EVP_CIPHER_CTX_iv_length(ctx.get())) ||
    !CBB_reserve(out, &ptr, session_len + EVP_MAX_BLOCK_LENGTH)) {
    return 0;
}
```



1. Analyze open-source implementations

- Ticket format
- Algorithms
- Look for immediate issues

2. Large-scale analysis

- Propose potential pitfalls
- How to gather tickets
- How to analyze tickets
- Perform Scan

Large Scale Evaluation

Potential Pitfalls

Offline

just need ticket

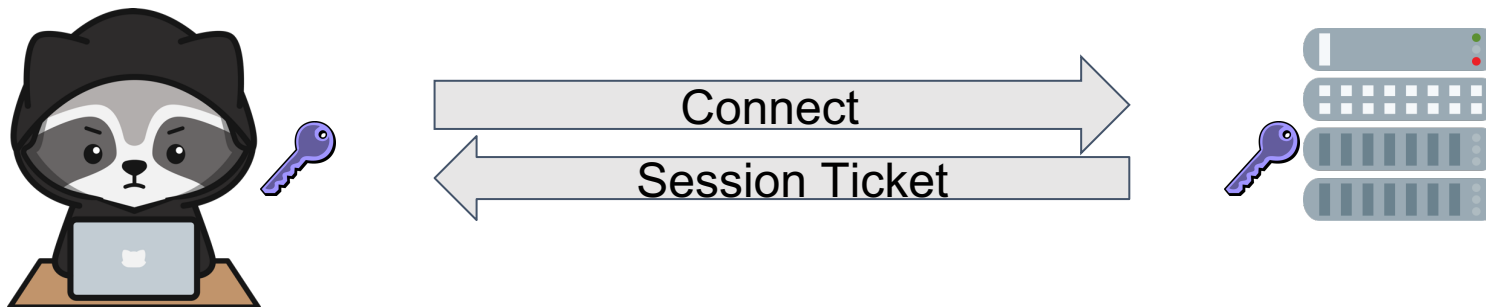
- Plaintext session tickets
 - Would still work
- Weak STEK (enc or auth)
 - 0x000000, 0x010203, ...
 - Also check weak algorithms
- Reused Keystream
 - e.g. CTR/GCM with reused nonce

Online

observe sever behavior

- Authentication Issues
 - Bitflips
 - CBC Padding Oracle

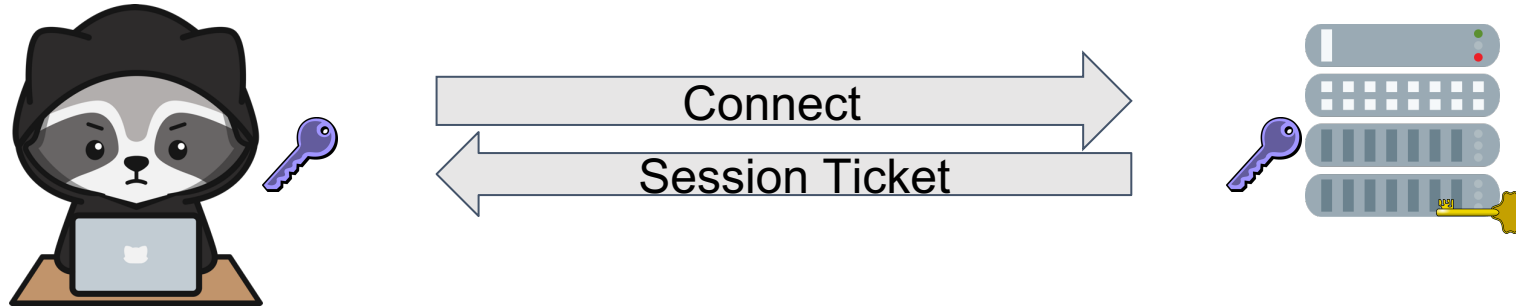
Large Scale Evaluation Plaintext Ticket



- Handshake Protocol: New Session Ticket
 - Handshake Type: New Session Ticket (4)
 - Length: 235
- TLS Session Ticket
 - Session Ticket Lifetime Hint: 100800 seconds (1 day, 4 hours)
 - Session Ticket Length: 229
 - Session Ticket: 02d32b3f673ba516106107535a8f78ad158f3134dca563e7...
- TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
- TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message
 - Content Type: Handshake (22)
 - Version: TLS 1.2 (0x0302)

0050	c0 00 e5 02 d3 2b 3f 67 3b a5 16 10 61 07 53 5a	...+?g ;...a.SZ
0060	8f 78 ad 15 8f 31 34 dc a5 63 e7 70 31 27 6d 73	.x...14. .c.p1'ms
0070	13 1c 6a 0c 09 bc b9 71 41 fd 60 fc 1a 4b dd 7d	.j...q A.`..K.}
0080	6d ac 0d 59 cb 45 48 c6 3e b0 02 3e 4f e7 f7 45	m..Y.EH. >..>0..E
0090	73 28 eb 73 93 58 7a c0 68 f7 c6 d4 d0 20 66 a7	s(.s.Xz. h... f.
00a0	77 8d 97 2e 0f f4 e6 2a d7 a3 59 db b2 a2 36 61	w.....* ..Y...6a
00b0	08 ae be 1b 61 d1 94 08 ee 1f 6a 64 35 79 8c 22	...a... ..jd5y."
00c0	a9 35 d8 7a 46 10 4f 87 22 67 9b d1 c5 f2 b6 16	.5.zF.0. "g.....

Large Scale Evaluation Testing for Weak STEK



- Handshake Protocol: New Session Ticket
 - Handshake Type: New Session Ticket (4)
 - Length: 235
- TLS Session Ticket
 - Session Ticket Lifetime Hint: 100800 seconds (1 day, 4 hours)
 - Session Ticket Length: 229

Session Ticket: 02d32b3f673ba516106107535a8f78ad158f3134dca563e7...


- TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
- TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message

Content Type: Handshake (22)

Version: TLS 1.2 (0x0302)

0050	c0 00 e5 02 d3 2b 3f 67 3b a5 16 10 61 07 53 5a	...+?g ;...a.SZ
0060	8f 78 ad 15 8f 31 34 dc a5 63 e7 70 31 27 6d 73	.x...14. .c.p1'ms
0070	13 1c 6a 0c 09 bc b9 71 41 fd 60 fc 1a 4b dd 7d	.j...q A.`..K.}
0080	6d ac 0d 59 cb 45 48 c6 3e b0 02 3e 4f e7 f7 45	m..Y.EH. >..>0..E
0090	73 28 eb 73 93 58 7a c0 68 f7 c6 d4 d0 20 66 a7	s(.s.Xz. h... f.
00a0	77 8d 97 2e 0f f4 e6 2a d7 a3 59 db b2 a2 36 61	w.....* ..Y...6a
00b0	08 ae be 1b 61 d1 94 08 ee 1f 6a 64 35 79 8c 22	...a... ..jd5y."
00c0	a9 35 d8 7a 46 10 4f 87 22 67 9b d1 c5 f2 b6 16	.5.zF.0. "g.....

Large Scale Evaluation Testing for Weak STEK


1. Decrypt Ticket
 2. Check for master secret  in plaintext
- Unknown algorithm, key, IV, structure

b26b38b49fa971dee53e5137b0e73226c74e0cbd21d5c4042fad0b0a8c6747ef5e43969a42ef883b1c529fc

IV

Ciphertext

Large Scale Evaluation Testing for Weak STEK


1. Decrypt Ticket
 2. Check for master secret  in plaintext
- Unknown algorithm, key, IV, structure

b26b38b49fa971dee53e5137b0e73226c74e0cbd21d5c4042fad0b0a8c6747ef5e43969a42ef883b1c529fc

IV

Ciphertext

Large Scale Evaluation Testing for Weak STEK


1. Decrypt Ticket
 2. Check for master secret  in plaintext
- Unknown algorithm, key, IV, structure

b26b38b49fa971dee53e5137b0e73226c74e0cbd21d5c4042fad0b0a8c6747ef5e43969a42ef883b1c529fc

IV

Ciphertext

Large Scale Evaluation Testing for Weak STEK

1. Decrypt Ticket
 2. Check for master secret  in plaintext
- Unknown algorithm, key, IV, structure

```
b26b38b49fa971dee53e5137b0e73226c74e0cbd21d5c4042fad0b0a8c6747ef5e43969a42ef883b1c529fc
```

IV

Ciphertext

- AES, ChaCha, DES, 3DES?
 - ECB, CBC, CTR, GCM?
- Key?
 - 0x000000, 0x000102, 0xFFFFFFFF

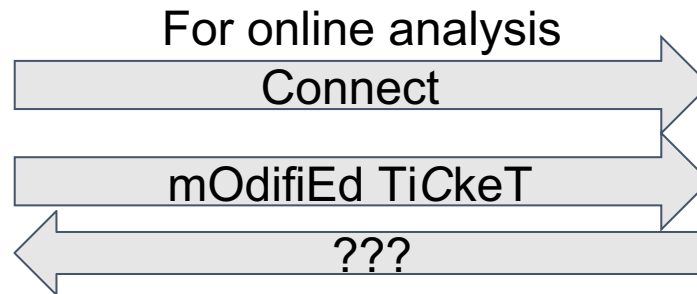
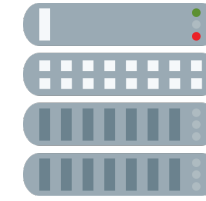
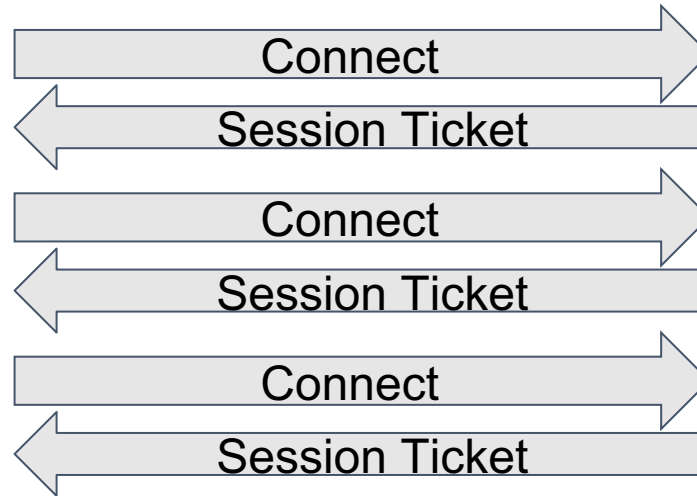
Large Scale Evaluation Brute Forcing the STEK

- Tested TLS “related” encryption algorithm combinations:
 - DES : ECB, CBC mode
 - 3DES : ECB, CBC mode
 - AES-128/256: ECB, CBC, CTR, CCM, GCM mode (ignoring auth)
 - ChaCha20
- Tested TLS “related” authentication algorithms:
 - HMAC: MD5, SHA1, SHA256, SHA384 and SHA512
- 144 potential keys
- >400 potential formats (encryption)

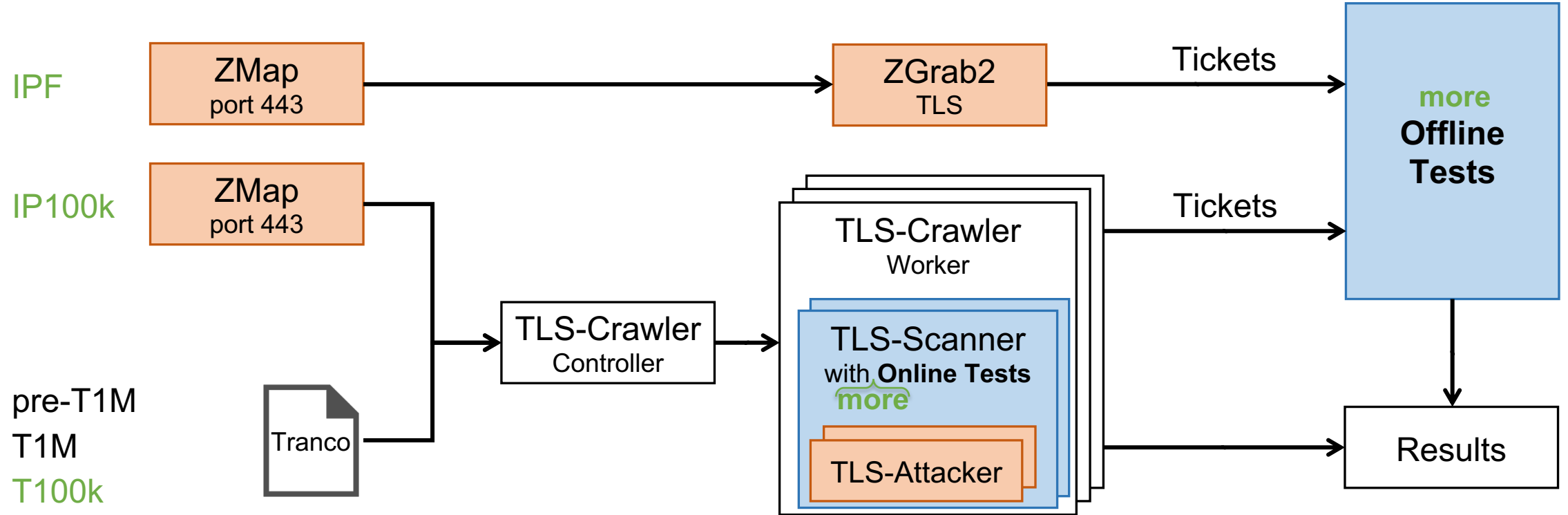
Large Scale Evaluation Methodology



10x TLS 1.3
10x TLS 1.2
...



Scans



Results

Scan	<u>Offline Analysis</u>			<u>Online Analysis</u>	
	Unencrypted Ticket	Weak STEK	Reused Keystream	Missing Auth. Protection	Padding Oracle
pre-T1M	0	1923	–	–	–

- Scope: Tranco 100k – **1.9%** vulnerable
- Most of the servers belonged to AWS
- STEK = 0x00 00 00...
- Reported April 2021, fixed within 8 hours
- Maybe introduced in September 2020 (internal NGINX change)

Results

Scan	Unencrypted Ticket	<u>Offline Analysis</u>		<u>Online Analysis</u>	
		Weak STEK	Reused Keystream	Missing Auth. Protection	Padding Oracle
pre-T1M	0	1923	–	–	–
T1M	0				
T100k	0				
IP100k	0				
IPF	0				

Results

Scan	<u>Offline Analysis</u>			<u>Online Analysis</u>	
	Unencrypted Ticket	Weak STEK	Reused Keystream	Missing Auth. Protection	Padding Oracle
pre-T1M	0	1923	–	–	–
T1M	0		–		
T100k	0		0		
IP100k	0		0		
IPF	0		1		

Results

Scan	<u>Offline Analysis</u>			<u>Online Analysis</u>	
	Unencrypted Ticket	Weak STEK	Reused Keystream	Missing Auth. Protection	Padding Oracle
pre-T1M	0	1923	–	–	–
T1M	0		–	–	–
T100k	0		0	0	0
IP100k	0		0	0	0
IPF	0		1	–	–

Results

Scan	<u>Offline Analysis</u>			<u>Online Analysis</u>	
	Unencrypted Ticket	Weak STEK	Reused Keystream	Missing Auth. Protection	Padding Oracle
pre-T1M	0	1923	–	–	–
T1M	0	3	–	–	–
T100k	0	1	0	0	0
IP100k	0	0	0	0	0
IPF	0	189	1	–	–

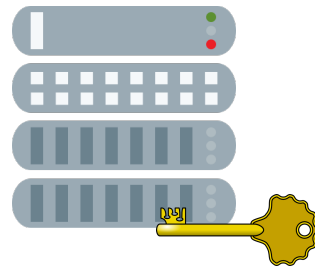
Results Summary

- No authentication issues
- One reused keystream
- Weak keys
 - Many 00-keys
 - 00 01 02 03 ...
 - Partially initialized keys
 - HMAC key initialized, AES key 0000
 - Half initialized

Encryption Key	Authentication Key
00 00 ... 00 00	-
00 00 ... 00 00	00 00 ... 00 00
10 11 ... 1e 1f	20 21 ... 2e 2f
31...31 00...00	31...31 00...00

Why Wasn't This Found Earlier?

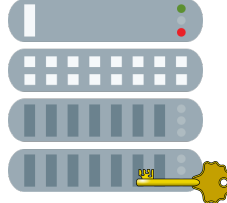
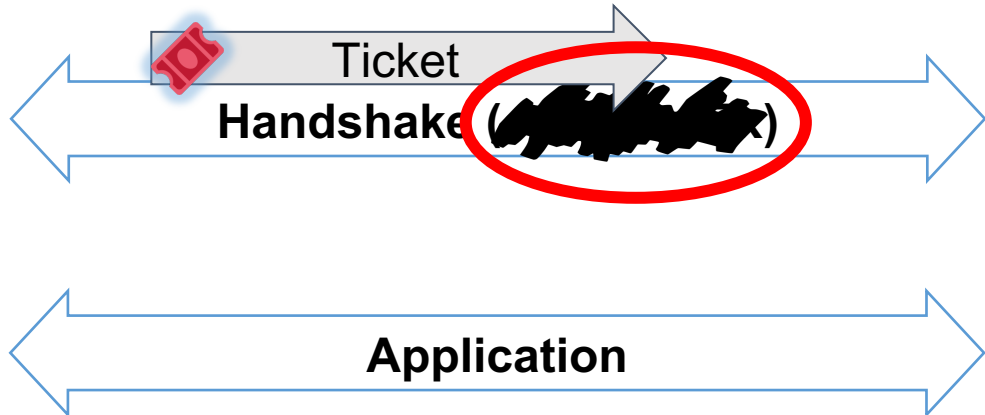
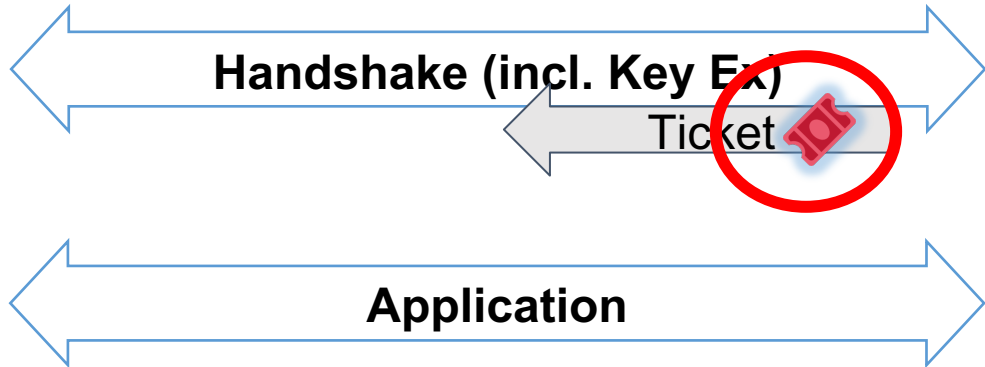
- Tickets still work
 - Including resumption
- STEK is hard to audit
 - Have to try each possible key
 - Ticket format unknown



- Handshake Protocol: New Session Ticket
 - Handshake Type: New Session Ticket (4)
 - Length: 235
- TLS Session Ticket
 - Session Ticket Lifetime Hint: 100800 seconds (1 day, 4 hours)
 - Session Ticket Length: 229
 - Session Ticket: 02d32b3f673ba516106107535a8f78ad158f3134dca563e7...
- TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
- TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message
 - Content Type: Handshake (22)
 - Version: TLS 1.2 (0x0302)

0050	c0 00 e5 02 d3 2b 3f 67 3b a5 16 10 61 07 53 5a	...+?g ;...a·SZ
0060	8f 78 ad 15 8f 31 34 dc a5 63 e7 70 31 27 6d 73	·x···14· ·c·p1'ms
0070	13 1c 6a 0c 09 bc b9 71 41 fd 60 fc 1a 4b dd 7d	··j····q A·`··K·}
0080	6d ac 0d 59 cb 45 48 c6 3e b0 02 3e 4f e7 f7 45	m··Y·EH· >··>0··E
0090	73 28 eb 73 93 58 7a c0 68 f7 c6 d4 d0 20 66 a7	s(·s·Xz· h···· f·
00a0	77 8d 97 2e 0f f4 e6 2a d7 a3 59 db b2 a2 36 61	w·····* ··Y···6a
00b0	08 ae be 1b 61 d1 94 08 ee 1f 6a 64 35 79 8c 22	····a···· ··jd5y·"
00c0	a9 35 d8 7a 46 10 4f 87 22 67 9b d1 c5 f2 b6 16	·5·zF·0· "g·····
00d0	7e 48 43 82 72 96 03 f6 8d bb cf dd 06 24 f4 9c	~HC·r·······\$.·
00e0	68 5b 4e 4d f0 a7 aa 06 ba 3b 31 56 f9 72 83 f9	h[NM····· ;1V·r··
00f0	92 f0 44 fe 19 25 44 7b 51 58 bc 24 9b 03 7e f0	··D·%D{ QX·\$.·~·
0100	13 0a 44 64 7f 06 b3 98 72 a4 7f 3f 89 14 3d 7c	··Dd····· r··?·..=
0110	ce cd 05 2c b1 0b 3d 47 89 d3 90 cb fa 03 5d ca	··,···=G ······].
0120	32 8b ef dd 44 1c cc 34 00 d1 0a 1a b4 46 4a 52	2···D···4 ·····FJR
0130	95 f2 60 43 9b a3 f5 b7 14 03 03 00 01 01 16 03	··`C····· ······

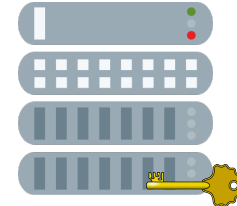
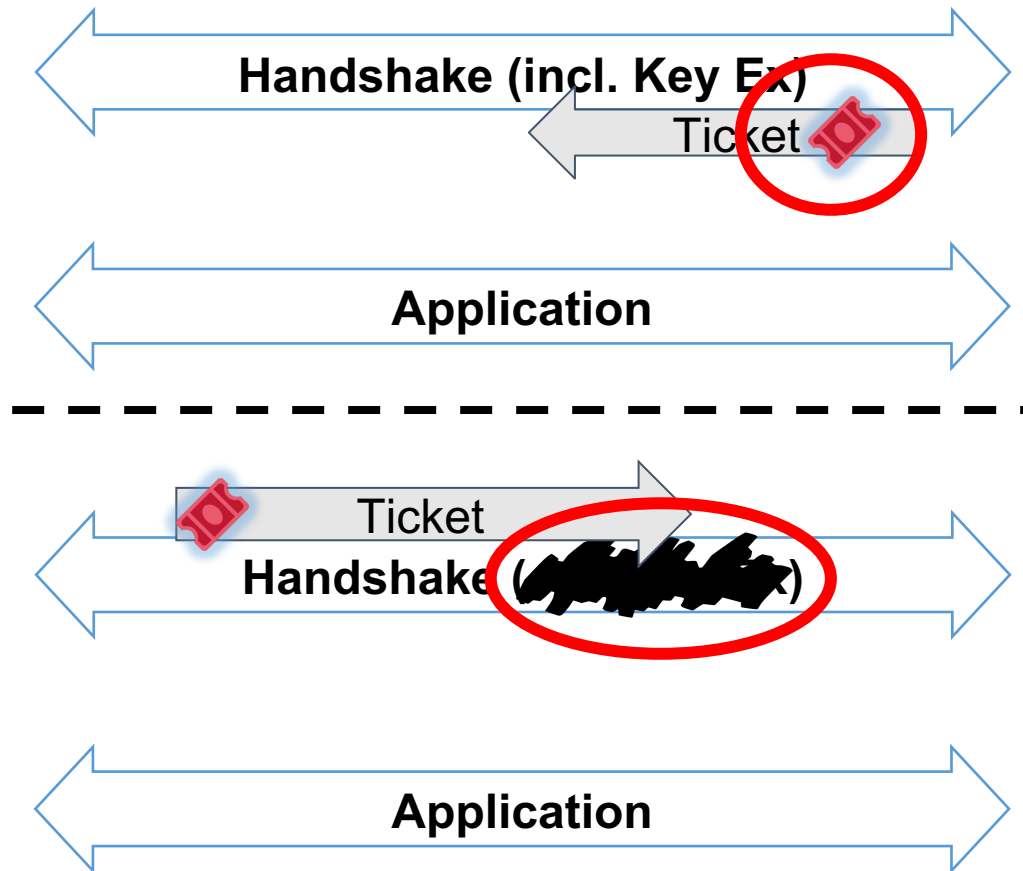
Issues in TLS 1.2



Issues:

- 1. No key exchange
- 2. Same secret reused
- 3. Tickets sent in plaintext

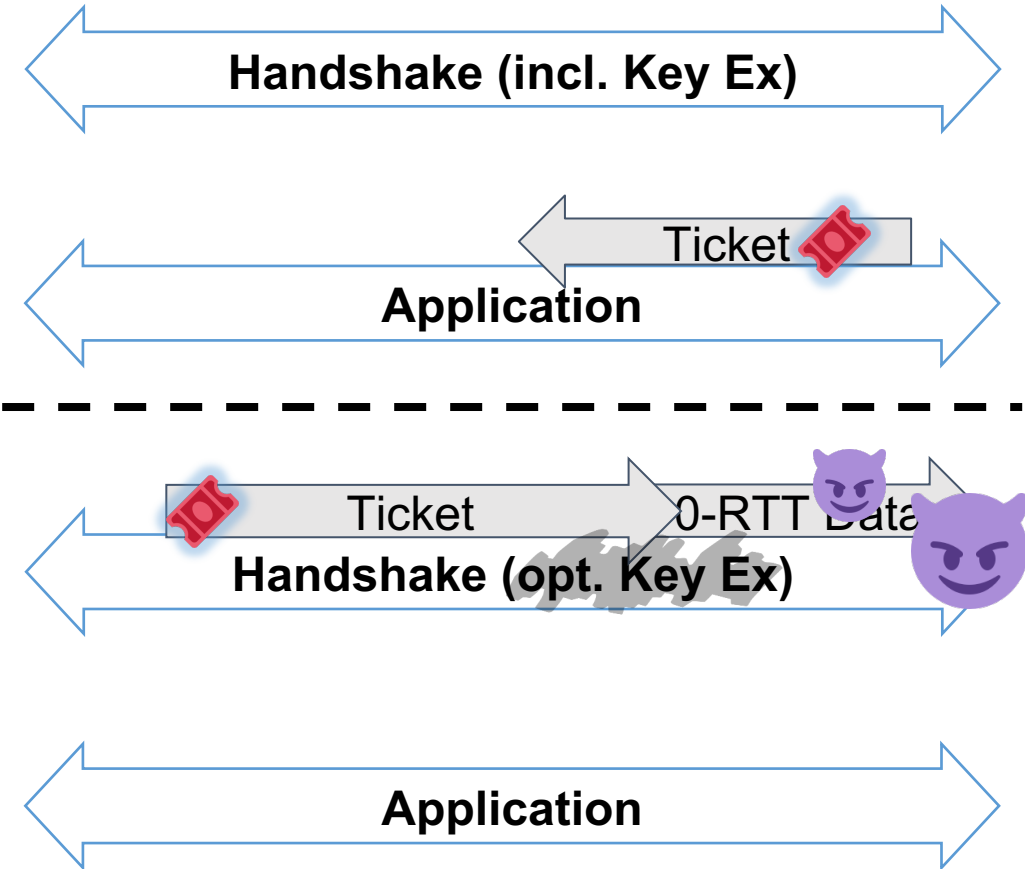
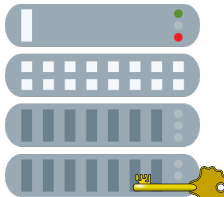
Improvements in TLS 1.3



Issues ➤ Solutions:

1. No key exchange
➤ Allow key exchange
2. Same secret reused
➤ Derive new secrets
3. Tickets sent in plaintext
➤ Sent encrypted

Issues in TLS 1.3



- If no Key Ex:
 - Decrypt Application
 - Read 0-RTT Data
 - Impersonate Server
- TLS 1.2 is widely used

We Really Need to Talk About Session Tickets



We Really Need(ed) to Talk About Session Tickets

Findings

- 0000 isn't a secure key
- Tickets undermine TLS security guarantees

Conclusions

- Hidden danger in:
 - Crypto shortcuts
 - Silently breaking crypto
 - Unauditable crypto

Takeaways

- Design protocols auditable for both parties
- Add defense in depth to your implementation
 - Check key material before use

We Really Need(ed) to Talk About Session Tickets

Findings

- 0000 isn't a secure key
- Tickets undermine TLS security guarantees

Conclusions

- Hidden danger in
 - Crypto shortcuts
 - Silently breaking crypto
 - Unauditable crypto

Takeaways

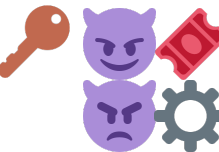
- Design protocols auditable for both parties
- Add defense in depth to your implementation
 - Check key material before use

Questions?

Attributions

 Based on MesserWoland, Crypto key, CC BY-SA 3.0
Path and Alignment slightly adapted, Colors changed for some figures

 Based on <https://publicdomainvectors.org/en/free-clipart/Vector-drawing-of-grayscale-key/31029.html>

 twemoji <https://github.com/twitter/twemoji>

 <https://publicdomainvectors.org/en/free-clipart/Vector-image-of-old-style-decorative-door-key/21178.html>

 <https://gitlab.com/rossel.jost/latex-twemojis/-/tree/master/src/twemojis-extra/>