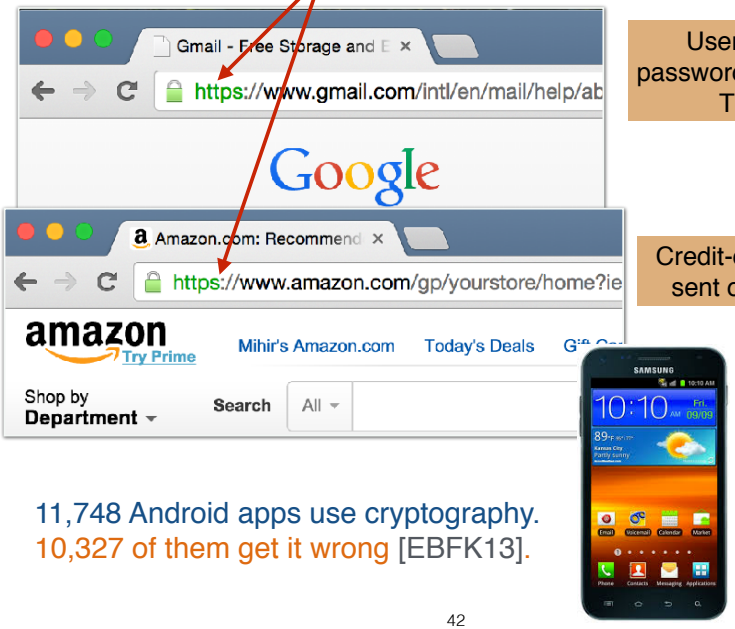
TLS/SSL

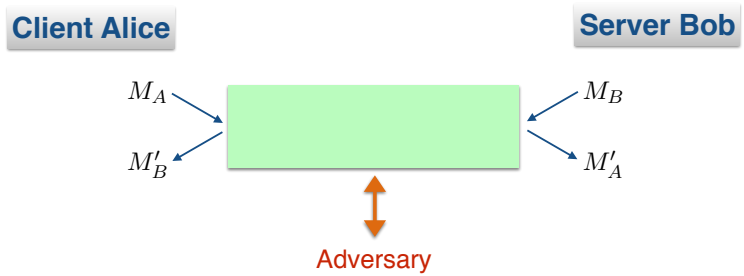


Username and password are sent over TLS/SSL

Credit-card number is sent over TLS/SSL

11,748 Android apps use cryptography.
10,327 of them get it wrong [EBFK13].

TLS/SSL aims to provide a secure channel

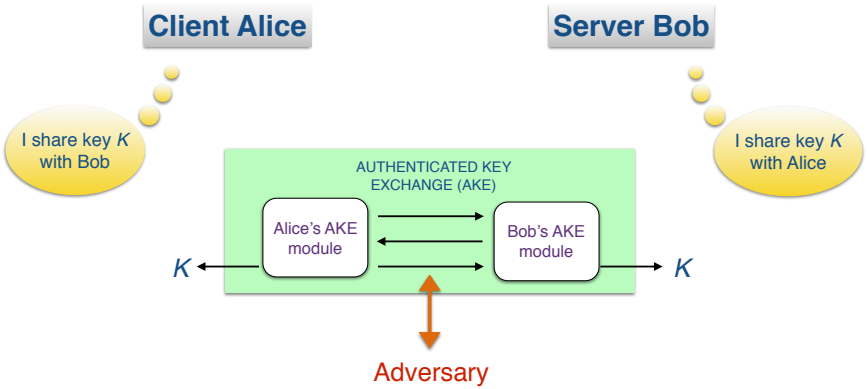


Privacy: Adversary does not learn anything about M_A, M_B

Authenticity: $M'_A = M_A$ and $M'_B = M_B$

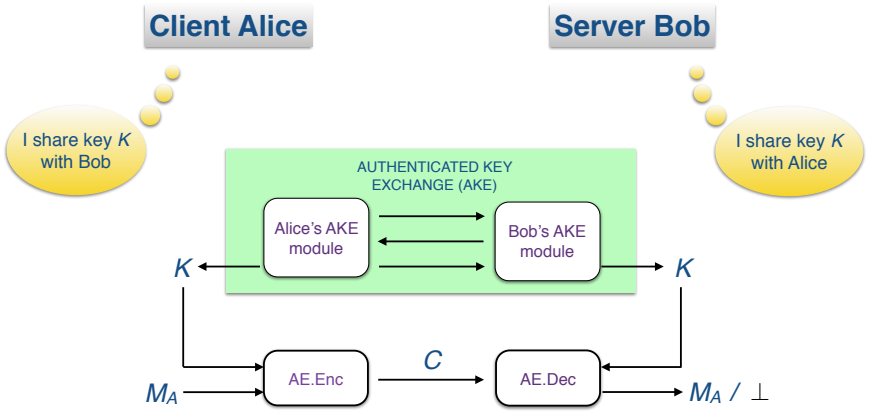
Identity: Alice is really ``Alice" and Bob is really ``Bob"

The cryptographic core of TLS/SSL



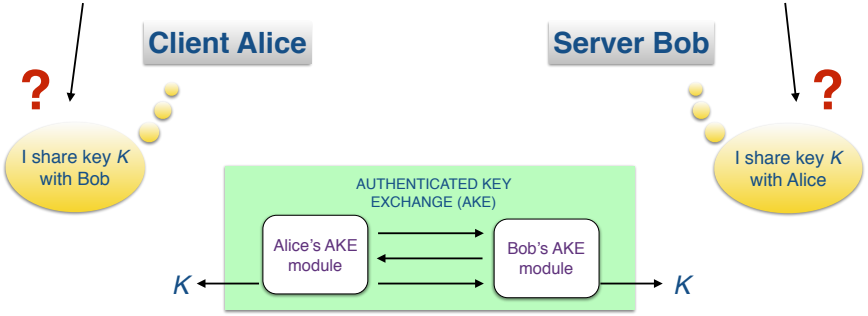
K is a fresh, authentic session key
Adversary cannot influence or know K

The cryptographic core of TLS/SSL



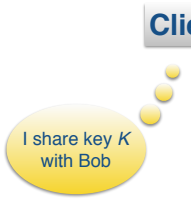
AE: Authenticated Encryption Scheme
C: Ciphertext
⊥: REJECT

The question of identity: Who are ``Alice" and ``Bob"?



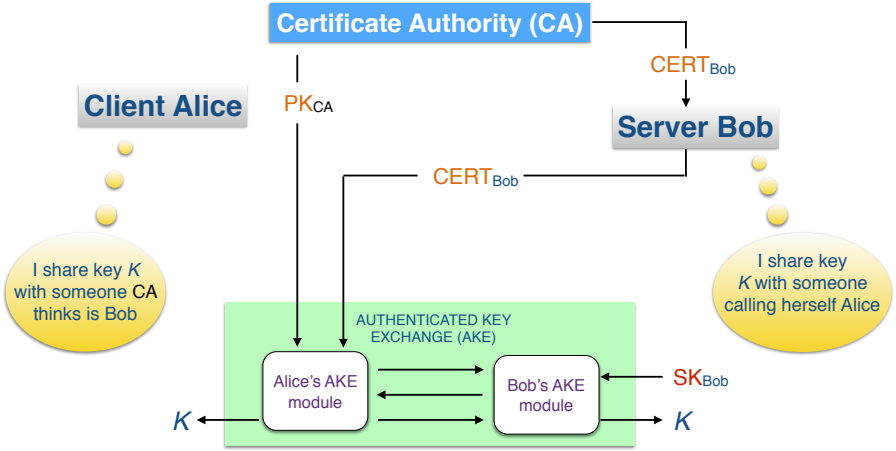
On the Internet, nobody knows you are a dog.

Who is ``Bob"?



google.com	Google
amazon.com	amazon
snapchat.com	Snapchat
twitter.com	

Think of Bob = example.com as a url or an ip address



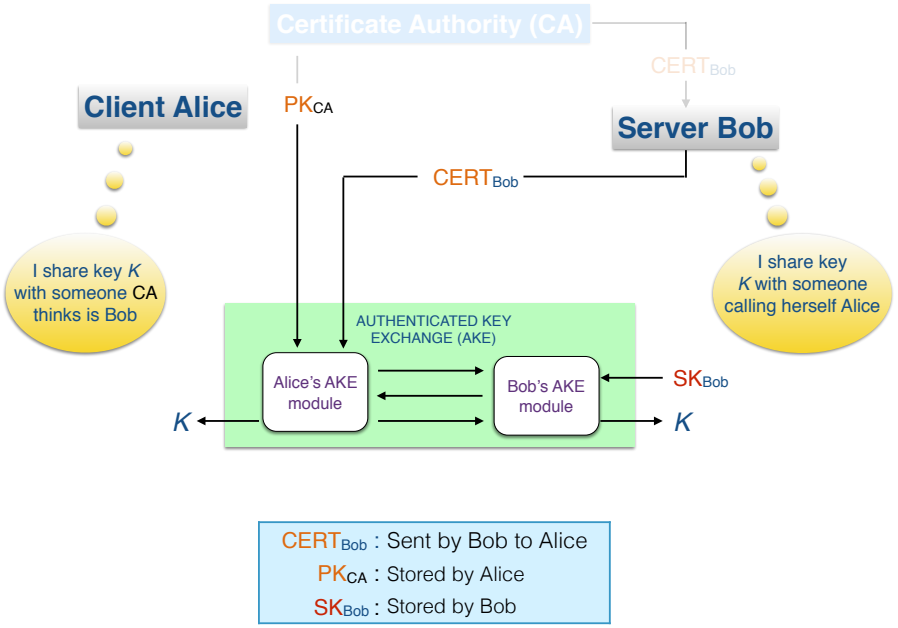
$CERT_{Bob}$: Bob's certificate, issued by CA
PK_{CA} : CA's public key
SK_{Bob} : Bob's secret key, tied to $CERT_{Bob}$

Certificate authorities

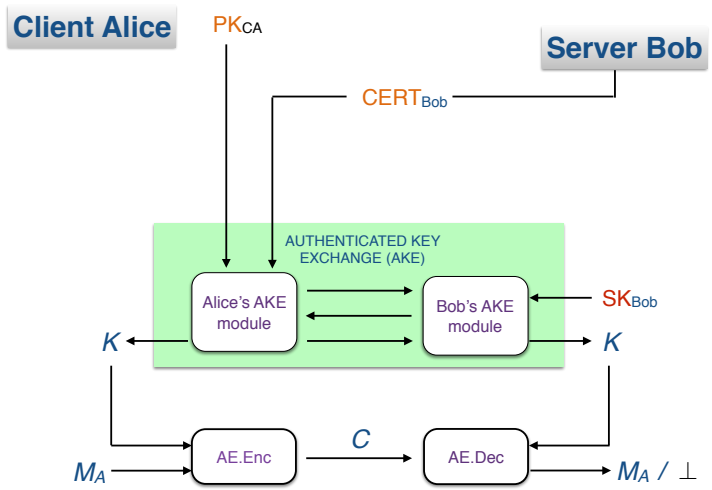
Rank	Issuer	Usage	market share
1.	Comodo	6.6%	33.6%
2.	Symantec Group	6.5%	33.2%
3.	Go Daddy Group	2.6%	13.2%
4.	GlobalSign	2.2%	11.3%
5.	DigiCert	0.6%	2.9%

As of February 2015

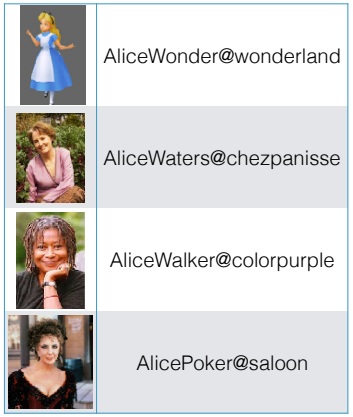
CA NAME	RATING
Comodo	(171 Reviews)
DigiCert	(1026 Reviews)
Entrust	(802 Reviews)
GeoTrust	(104 Reviews)
GlobalSign	(185 Reviews)
GoDaddy	(98 Reviews)
Network Solutions	(12 Reviews)
SSL.com	(44 Reviews)
StartCom	(330 Reviews)
SwissSign	(4 Reviews)

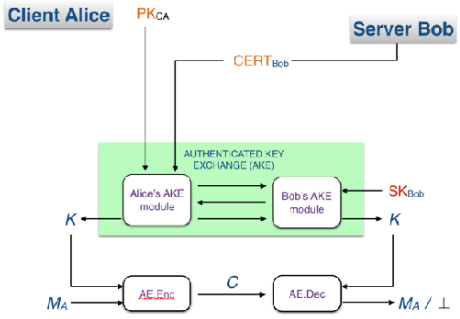


The cryptographic core of TLS/SSL



But who is ``Alice?"



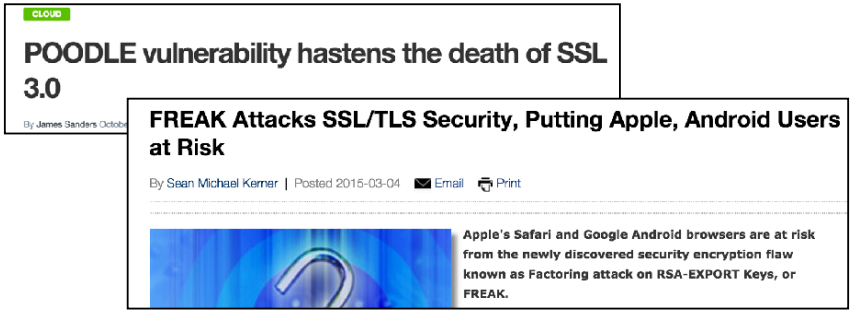


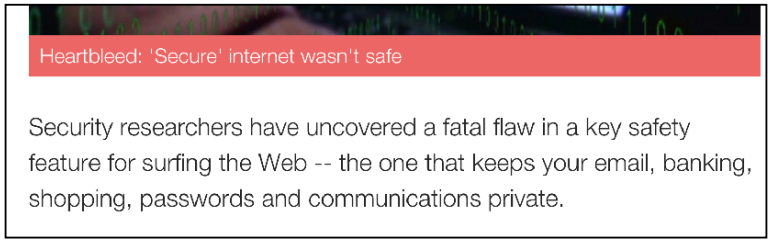
Default TLS/SSL provides **unilateral authentication**: Bob authenticates himself to Alice but not vice versa.

Alice does not typically have a certificate.

Alice will typically authenticate herself to Bob with username and password over the TLS/SSL channel itself.

TLS/SSL Vulnerabilities





TLS/SSL Vulnerabilities

Vulnerability	crypto	Implementation/ Usage
FREAK	x	
Re-negotiation	x	
Version Rollback		x
BEAST	x	
Padding Oracle	x	
Lucky 13	x	
Poodle	x	x
Heartbleed		x
RC4	x	
AllYourSSLsAreBelongToUs		x

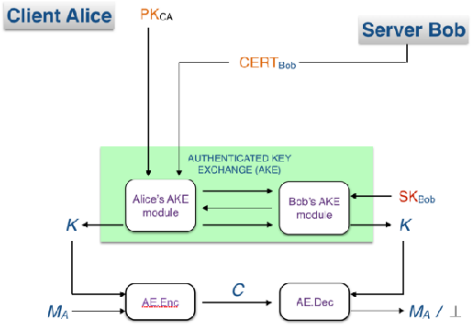
Many different TLS/SSL Implementations: OpenSSL, GnuTLS, cryptlib, JSSE, RSA BSafe, SChannel, …

Issues: Cipher suites, re-negotiation, side-channels, buffer overflows, bad randomness, …

Lots of bad crypto in TLS/SSL, often for historic and legacy reasons.

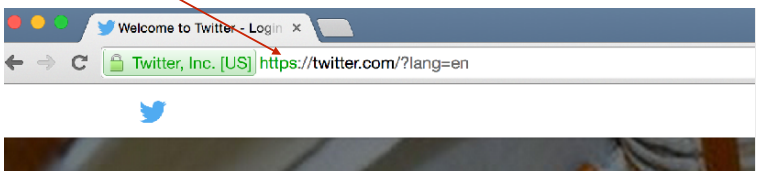
Get it right!

The cryptographic core of a secure channel



Summary, take away

TLS/SSL: Appreciate that there is a ton going on every time you access a website!



Providing a well-designed and analyzed cryptographic core is a central problem for us to address.