

Attacks in practice in cryptography

Paul Grubbs
ASCrypto 2023

paulgrub@umich.edu
@pag_crypto

1

About Me

- Assistant professor at University of Michigan
- Research in applied cryptography
- Did crypto in industry before grad school
- 5-month-old son, born in May
- Licensed Extra-class amateur radio operator
 - Call sign KE8WII
- First time in Ecuador



2

About this talk

- Violate security guarantees in threat model



- Secrecy, integrity, authenticity
- Anonymity, deniability, committing
- Not usually: availability, consistency, etc.



Who are the entities?
What capabilities do they have?

Use case studies of my own research to answer these questions:

1. Why can real-world cryptography *be* attacked?
2. How is real-world cryptography attacked?
3. Why do research on attacks?
4. Is attack research scientifically valuable?

3

Why are attacks possible?

- Provable security: rigorous way of ruling out attacks

Theorems say:

- If problem X is hard, then scheme Y is secure
- If scheme Y is secure, then protocol Z is secure

- Important distinction: attack on paper vs. system
- Security proofs don't apply to systems!
- Several ways attacks can happen:
 - Implementation of cryptography is wrong
 - Wrong cryptography is used
 - System "surrounding" cryptography doesn't provide the right guarantees

4

Why do attacks research?

- Need to find vulnerabilities before hackers do
- It's fun 😊
- Occasionally get paid (bug bounties)
- Publish papers

5

Are attacks scientifically valuable?

- Common criticism: "implementor just made a mistake"
- True sometimes, but implementor mistakes often point to deeper issues
- Provably-security implications
 - Proofs are wrong/vague/underspecified
 - Proofs don't rule out some attack
 - Proof does not apply to deployed crypto (use wrong primitives to instantiate)
 - Meaningful attacks go outside 'model' of proof.
 - May need new/better models!
- Cryptography design implications
 - Need schemes that are hard to use incorrectly
 - Invalid curve attacks, authenticated encryption, small subgroups, ...

6

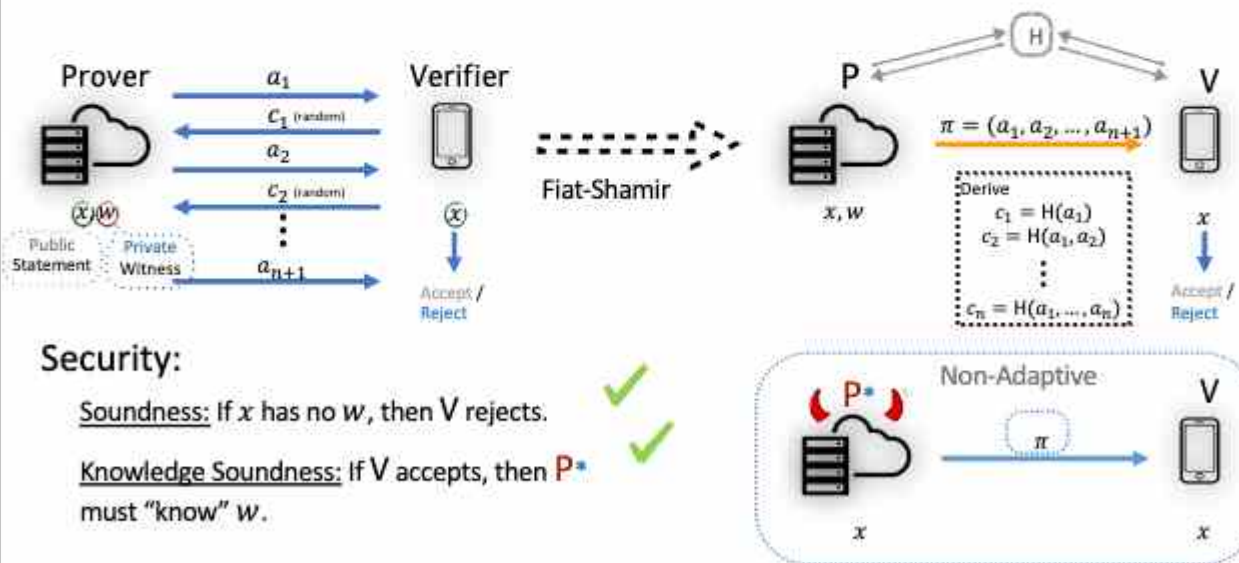
Talk Outline

Three case studies:

1. Weak Fiat-Shamir Attacks on Modern Proof Systems
 Dao, Miller, Wright, G. (IEEE S&P '23)
2. Attack on Facebook's Message Franking protocol
 Dodis, G., Ristenpart, Woodage (CRYPTO '18)
3. Why Your Encrypted Database Is Not Secure
 G., Ristenpart, Shmatikov (HotOS '17)

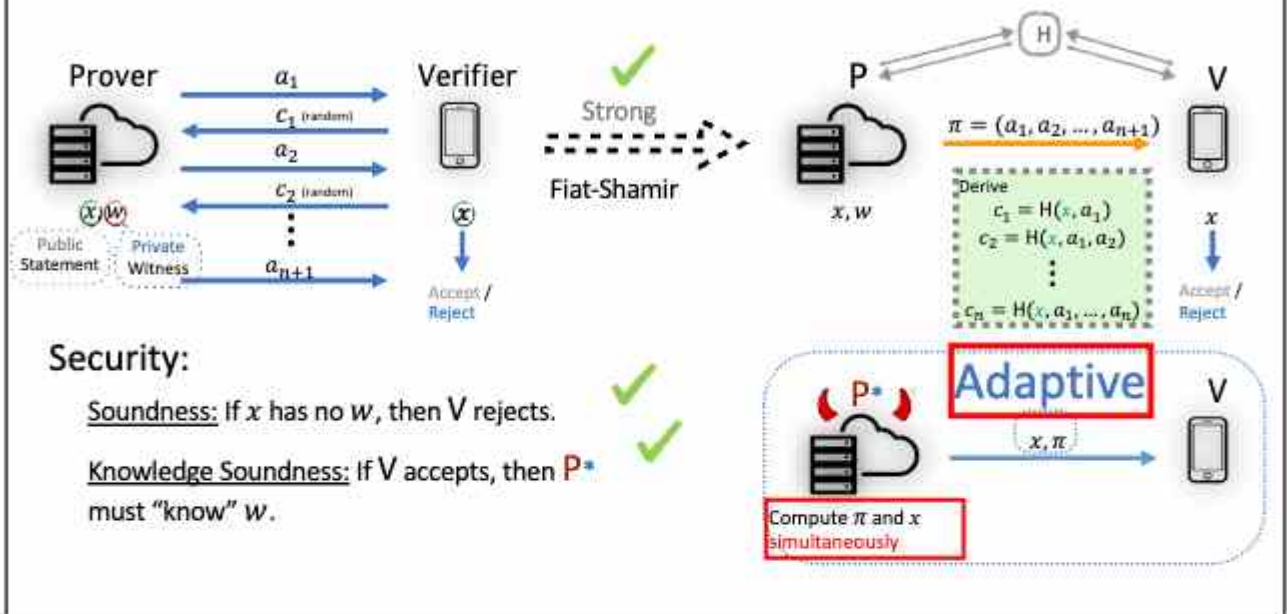
7

Proof Systems from Fiat-Shamir



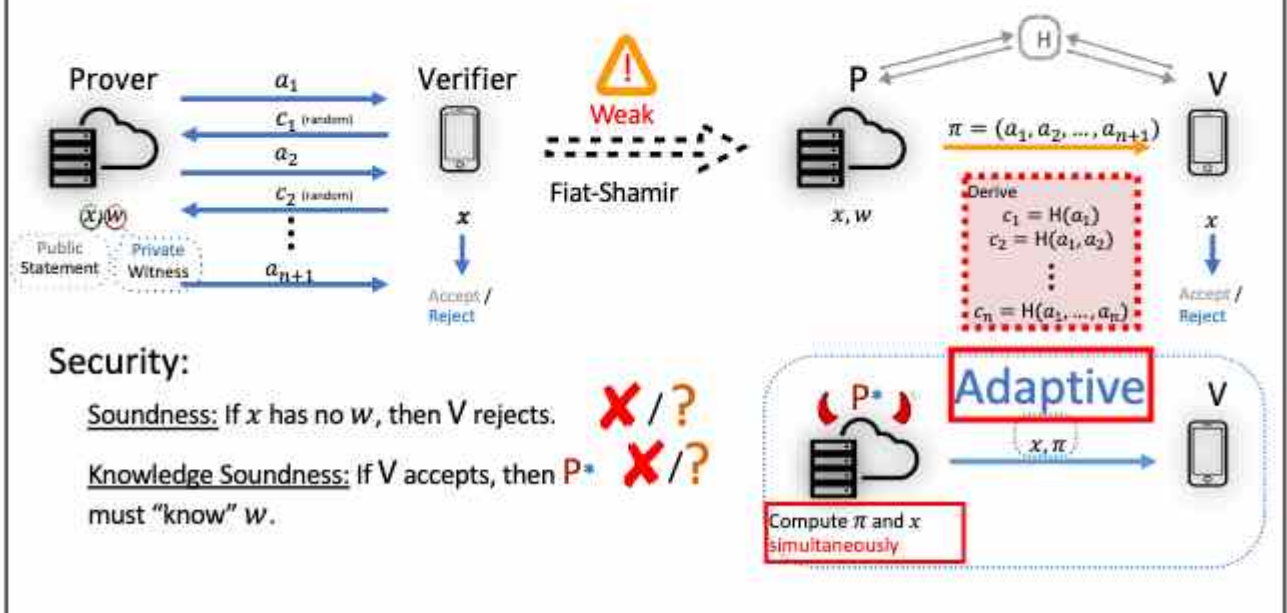
8

Strong Fiat-Shamir for Adaptive Security



9

Weak Fiat-Shamir and Attacks




10

Weak Fiat-Shamir and Attacks

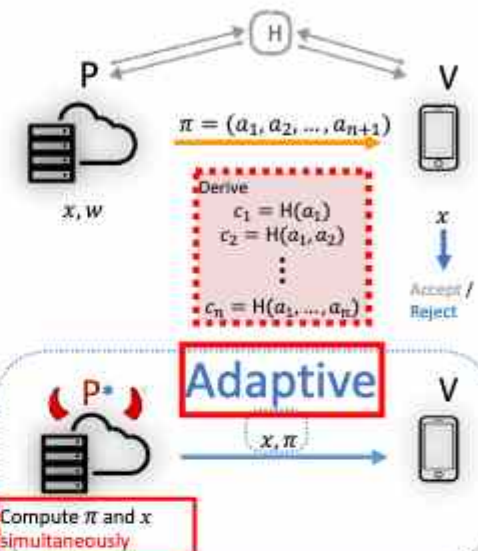
Simple Σ -Protocols
(e.g. Schnorr)

Statement $\xrightarrow{\text{Witness } a_{n+1}}$



Weak

Fiat-Shamir



Adaptive

Compute π and x simultaneously

**How not to Prove Yourself:
Pitfalls of the Fiat-Shamir Heuristic and
Applications to Helios**

David Bernhard¹, Olivier Pereira², and Bogdan Warinschi¹

How not to prove your election outcome


Thomas Hainz¹, Sarah Jamie Lewis¹, Olivier Pereira², and Vinusca Teague¹

11

Weak Fiat-Shamir and Attacks

Simple Σ -Protocols
(e.g. Schnorr)

Statement $\xrightarrow{\text{Witness } a_{n+1}}$



Weak

Fiat-Shamir

Modern Proof Systems
(e.g. Bulletproofs)

$c_n = H(a_1, \dots, a_n)$

Accept / Reject

How not to Prove Yourself:

1. Are there Weak Fiat-Shamir Attacks against Modern Proof Systems?
2. Do Modern-Day Systems Implement Weak Fiat-Shamir?
3. How Severe are Weak Fiat-Shamir Vulnerabilities?

Thomas Hainz¹, Sarah Jamie Lewis¹, Olivier Pereira², and Vinusca Teague¹

12

Results

- Survey of 75+ open-source implementations:
36 weak F-S vulnerabilities across 12 different proof systems.
- Explicit Attacks against Bulletproofs, Plonk, Spartan, and Wesolowski's VDF:
Provably break adaptive (knowledge) soundness.
- Case Studies of Practical Impacts:
 Create unlimited currency in two blockchain protocols.

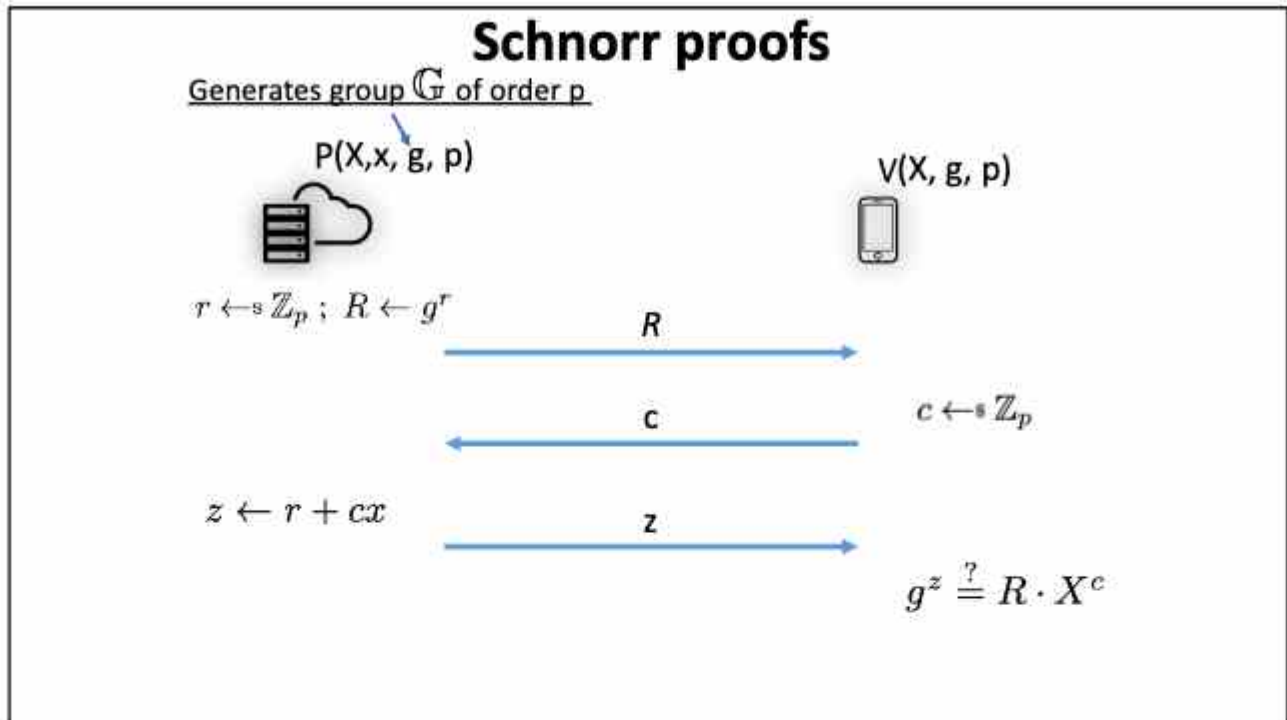
Proof System	Collision	Weak F-S?	Proof System	Collision	Weak F-S?
Bulletproofs [22]	liger [47]	✓	Plonk [37]	marco-polo [31]	✓
	bulletproofs [2]	✓		gnoll [32]	✓
	simple-bulletproofs [35]	✓		darknetwork [17]	✓
	bulletproofs0 [50]	✓		crab [38]	✓
	plonk-bulletproof [39]	✓		ZK-Gargo [39]	✓
	spartan-bulletproof [3]	✓		plinky [37]	✓
	blonk [49]	✓		okp-ep [34]	✓
	weppro-ibon [11]	✓		haki [45]	✓
	omni-bulletproof [13]	✓		of-ibn [71]	✓
	Zerify-X [48]	✓		okp-ibn [34]	✓
	okp [52]	✓		okp-ibn [34]	✓
	okp-ep [40]	✓		okp-ibn [34]	✓
	bulletproofs [23]	✓		okp-ibn [34]	✓
	omni [49]	✓		okp-ibn [34]	✓
dark-bulletproofs [25]	✓	okp-ibn [34]	✓		
okp2561-ibn [73]	✓	okp-ibn [34]	✓		
bulletproofs-zero [74]	✓	okp-ibn [34]	✓		
bulletproof [43]	✓	okp-ibn [34]	✓		
blonk [49]	✓	okp-ibn [34]	✓		
ibn [44]	✓	okp-ibn [34]	✓		
Bulletproofs maroc [41]	dark-bulletproofs [25]	✓	Spartan [82]	spartan [34]	✓
	ep-bonus [41]	✓		okp-ep [34]	✓
Sonic [42]	okp-ibn [34]	✓	Luna [32]	okp-ep [34]	✓
	okp-ibn [34]	✓		okp-ep [34]	✓
	okp-ibn [34]	✓		okp-ep [34]	✓
Sonic [42]	okp-ibn [34]	✓	Bokubean [15]	okp-ep [34]	✓
	okp-ibn [34]	✓		okp-ep [34]	✓
Sonic [42]	okp-ibn [34]	✓	Nova [32]	okp-ep [34]	✓
	okp-ibn [34]	✓		okp-ep [34]	✓
Sonic [42]	okp-ibn [34]	✓	Graft [37]	okp-ep [34]	✓
	okp-ibn [34]	✓		okp-ep [34]	✓

13

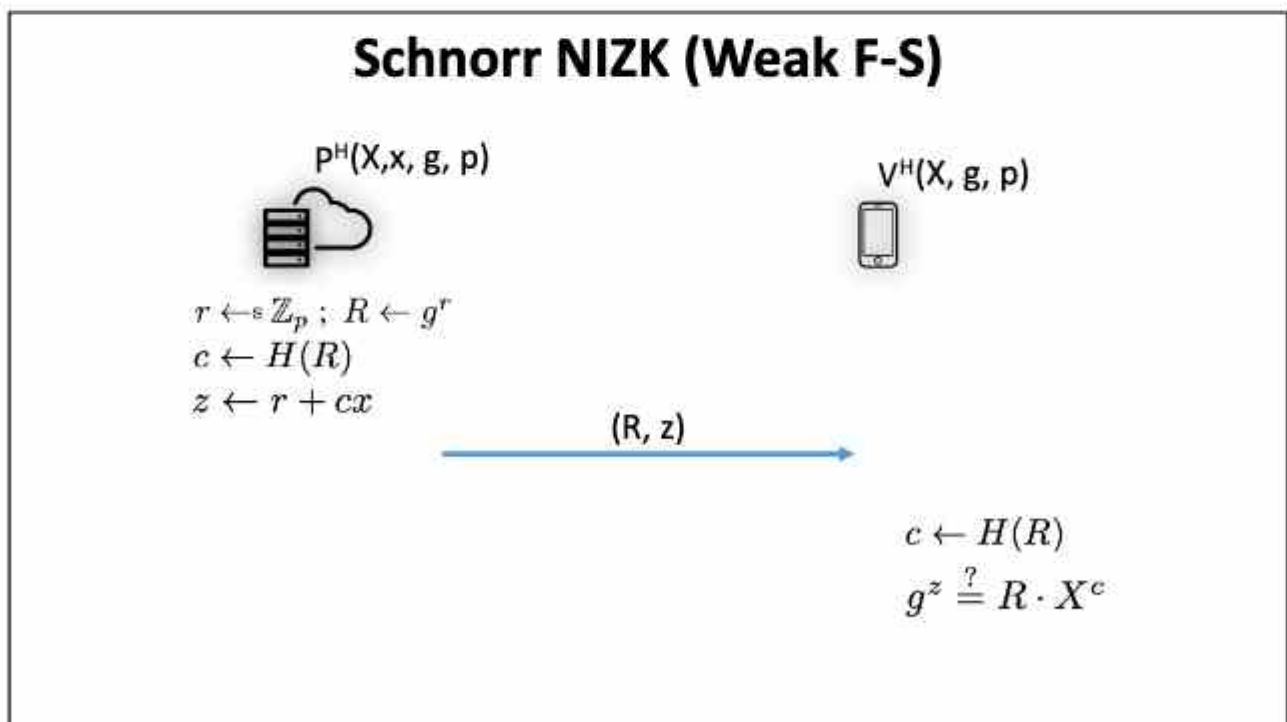
Weak Fiat-Shamir Attacks

(as easy as solving a linear equation)

14

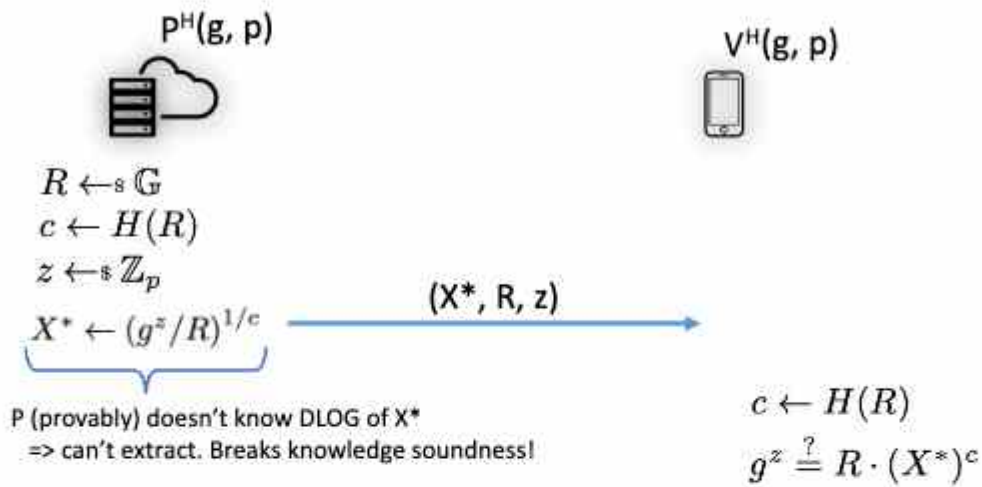


15



16

Adaptive Attack on Weak F-S Schnorr



[BPW12]

17

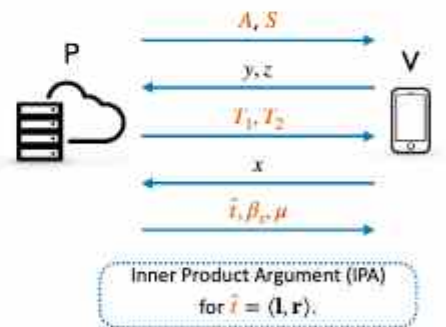
Bulletproofs - Protocol Description

Aggregate Range Proof Relation:

- $V_1 = g^{v_1} h^{t_1}, \dots, V_m = g^{v_m} h^{t_m}$
- $v_1, \dots, v_m \in [0, 2^n - 1]$

Note: $T_1 = g^{t_1} h^{\beta_1}, T_2 = g^{t_2} h^{\beta_2}$ in an honest proof

(with $t_1, \beta_1, t_2, \beta_2$ known by P)



(along with IPA check)

18

Bulletproofs - Weak Fiat-Shamir Attack

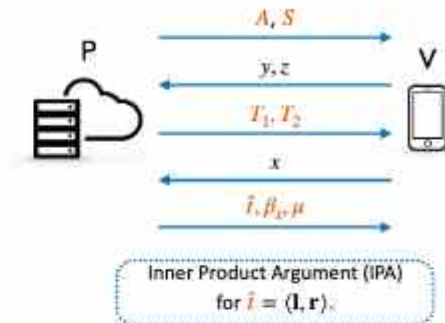
Aggregate Range Proof Relation:

- $V_1 = g^{v_1} h^{r_1}, \dots, V_m = g^{v_m} h^{r_m}$
- $v_1, \dots, v_m \in [0, 2^n - 1]$

Weak F-S Attack: When V_1, \dots, V_m are not hashed

1. Compute P's messages using an *arbitrary* witness:
 - Set $T_1 = g^{t_1} h^{\beta_1}, T_2 = g^{t_2} h^{\beta_2}$ for *arbitrary* $t_1, t_2, \beta_1, \beta_2$.
2. Solve for $v_1, \dots, v_m, \gamma_1, \dots, \gamma_m$ that satisfy (1).

$$\begin{cases} v_1 z^2 + \dots + v_m z^{m+1} = \hat{t} - \delta(y, z) - t_1 x - t_2 x^2 \\ \gamma_1 z^2 + \dots + \gamma_m z^{m+1} = \beta_x - \beta_1 x - \beta_2 x^2 \end{cases} \quad (1)$$



19

Practical Impacts

21

Overview of Vulnerable Implementations

Which projects are vulnerable?

- Most are clearly: proof of concepts, academic projects, or marked as non-production ready.
- Most implementations in production are not affected

Bulletproofs [22]	br-go [383]	✓
	bulletproof-js [17]	✓
	simple-bulletproof-js [344]	✓
	BulletproofSmith [20]	✓
	python-bulletproofs [79]	✓
	adjust-bulletproofs [7]	✓
	rlc-go [199]	✓
	incognito-chain [52]	✓*
	incognito-bulletproofs [33]	✓*
	ZenGo-X [97]	✓*
	rlcjs [55]	✓*
	rlc-rlp [82]	✓*
	bulletproofrb [21]	✓*
	incognito [89]	✗
	dash-bulletproofs [79]	✗
	secp256k1-rlp [76]	✗
bulletproofs-ocaml [75]	✗	
tail-project [86]	✗	
Incognito [90]	✗	
Grin [44]	✗	
Bulletproofs variant [40]	dash-bulletproofs [79]	✓*
	cpp-ibwero [63]	✗

22

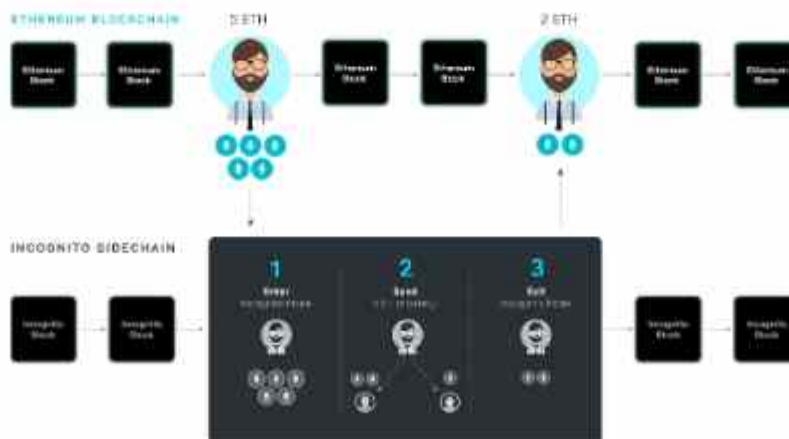
Case Study: Incognito Chain

Description:

○ Incognito

The privacy layer of crypto

\$250M+ VALUE SHIELDED **+6M** ANONYMOUS TRANSACTIONS **100+** CORE DEVELOPERS **16** BLOCKS SUPPORTED



23

Case Study: Incognito Chain

Description: ○ Incognito **The privacy layer of crypto**

Proof Relation:

- **Equality check:** $\sum v_{in} = \sum v_{out}$ ⇐ enforced by (linkable) ring signature
- **Range check:** $v_{in}, v_{out} \in [0, 2^{64} - 1], \forall$ input & output ⇐ enforced by BP aggregate range proofs

Weak F-S Attack:

- Choose $\vec{v}_{in}, \vec{v}_{out}$ to satisfy equality check as well as BP verification equation

➔

$$\begin{cases} v_1 = v_2 + v_3 + v_4 \\ v_1 z^2 + v_2 z^3 + v_3 z^4 + v_4 z^5 = \hat{t} - \delta(y, z) - t_1 x - t_2 x^2 \\ \gamma_1 z^2 + \gamma_2 z^3 + \gamma_3 z^4 + \gamma_4 z^5 = \beta_x - \beta_1 x - \beta_2 x^2 \end{cases}$$

(input v_1 and outputs v_2, v_3, v_4)

24

Case Study: Incognito Chain

Description: ○ Incognito **The privacy layer of crypto**

Proof Relation:

- **Equality check:** $\sum v_{in} = \sum v_{out}$ ⇐ enforced by (linkable) ring signature
- **Range check:** $v_{in}, v_{out} \in [0, 2^{64} - 1], \forall$ input & output ⇐ enforced by BP aggregate range proofs

Weak F-S Attack:

- Choose $\vec{v}_{in}, \vec{v}_{out}$ to satisfy equality check as well as BP verification equation

1 PRV

1 bazillion PRV!

➔

$$\begin{cases} v_1 = v_2 + v_3 + v_4 \\ v_1 z^2 + v_2 z^3 + v_3 z^4 + v_4 z^5 = \hat{t} - \delta(y, z) - t_1 x - t_2 x^2 \\ \gamma_1 z^2 + \gamma_2 z^3 + \gamma_3 z^4 + \gamma_4 z^5 = \beta_x - \beta_1 x - \beta_2 x^2 \end{cases}$$

(input v_1 and outputs v_2, v_3, v_4)

25

Case Study: Incognito Chain

Description: ○ Incognito The privacy layer of crypto

Proof Relation:

- **Equality check:** $\sum v_{in} = \sum v_{out}$ ← enforced by (linkable) ring signature
- **Range check:** $v_{in}, v_{out} \in [0, 2^{64} - 1], \forall$ input & output ← enforced by ~~linkable~~ aggregate range proofs

Weak F-S Attack:

- Choose $\vec{v}_{in}, \vec{v}_{out}$ to satisfy equality check as well as BP verification equation

1 PRV →

$$\begin{cases} v_1 = v_2 + v_3 + v_4 & \text{1 bazillion PRV!} \\ v_1 z^2 + v_2 z^3 + v_3 z^4 + v_4 z^5 = \hat{t} - \delta(y, z) - t_1 x - t_2 x^2 \\ \gamma_1 z^2 + \gamma_2 z^3 + \gamma_3 z^4 + \gamma_4 z^5 = \beta_x - \beta_1 x - \beta_2 x^2 \end{cases}$$

(input v_1 and outputs v_2, v_3, v_4)

26

Why is there so much weak F-S?

27

Insufficient Coverage of “correct” Fiat-Shamir

How is Fiat-Shamir presented in academic papers?

1. Mention that Fiat-Shamir can be applied, with no specification for the transform.

Removing interaction. Our construction can be made non-interactive in the random oracle model using Fiat-Shamir heuristic [29]. Though GKR protocol is not constant round, recent results [14, 27] show that as well. Finally, public-coin interactive arguments may be cryptographically compiled into SNARKs using the Fiat-Shamir transform.

subsequent step, the argument can be made non-interactive via the Fiat-Shamir transformation, and thereby obtain a preprocessing SNARG with universal SRS.

We apply the Fiat-Shamir heuristic to the protocol from Section 3 to obtain a non-interactive argument of knowledge that is secure in the random oracle model

Hyrax-I is a public-coin protocol, we apply the Fiat-Shamir heuristic [45] to produce a zkSNARK that we call Hyrax whose

Challenges are random field elements. In practice we assume that the Fiat-Shamir heuristic would be applied in order to obtain a non-interactive argument of knowledge that is secure in the random oracle model.

The above SNARK is obtained via a popular paradigm that combines a polynomial IOP and a polynomial commitment scheme in order to obtain an interactive argument, and then relies on the Fiat-Shamir paradigm

Finally, since our protocol is public coin, it can be made non-interactive in the random oracle model using the Fiat-Shamir transform [53], thereby obtaining a family

be made non-interactive in the random oracle model using the Fiat-Shamir transform [FSS0], and be instantiated (heuristically) in the plain model using a

witness-extended emulation. Applying the Fiat-Shamir transform [FSS0] to the public-coin interactive argument results in the claimed SNARK for \mathcal{R}_{GKR} .

28

Insufficient Coverage of “correct” Fiat-Shamir

How is Fiat-Shamir presented in academic papers?

1. Mention that Fiat-Shamir can be applied, with no specification for the transform.
2. Attempt to specify Fiat-Shamir:
 - ⇒ (some) do not get it right on the first try!

Plonk:

Compute quotient challenge $\alpha \in \mathbb{F}_p$:

$$\alpha = H([a], [b], [c], [z])$$

(December 2019)



We describe the protocol below as a non-interactive protocol using the Fiat-Shamir heuristic. For this purpose we always denote by transcript the concatenation of the common preprocessed input, and public input, and the proof elements written by the prover up to a certain point in time. We use transcript for obtaining random challenges via

(March 2020)

29

Insufficient Coverage of “correct” Fiat-Shamir

How is Fiat-Shamir presented in academic papers?

1. Mention that Fiat-Shamir can be applied, with no specification for the transform.
2. Attempt to specify Fiat-Shamir:
 - ⇒ (some) do not get it right on the first try!

random challenges are replaced by hashes of the transcript up to that point. For instance $y = H(A, S)$ and $z = H(A, S, y)$ (July 2018)

Bulletproofs:

random challenges are replaced by hashes of the transcript up to that point, including the statement itself. For example, one could set $y = H(st, A, S)$ and $z = H(A, S, y)$, where st is the statement. (April 2022)

(in response to our FrozenHeart disclosure)

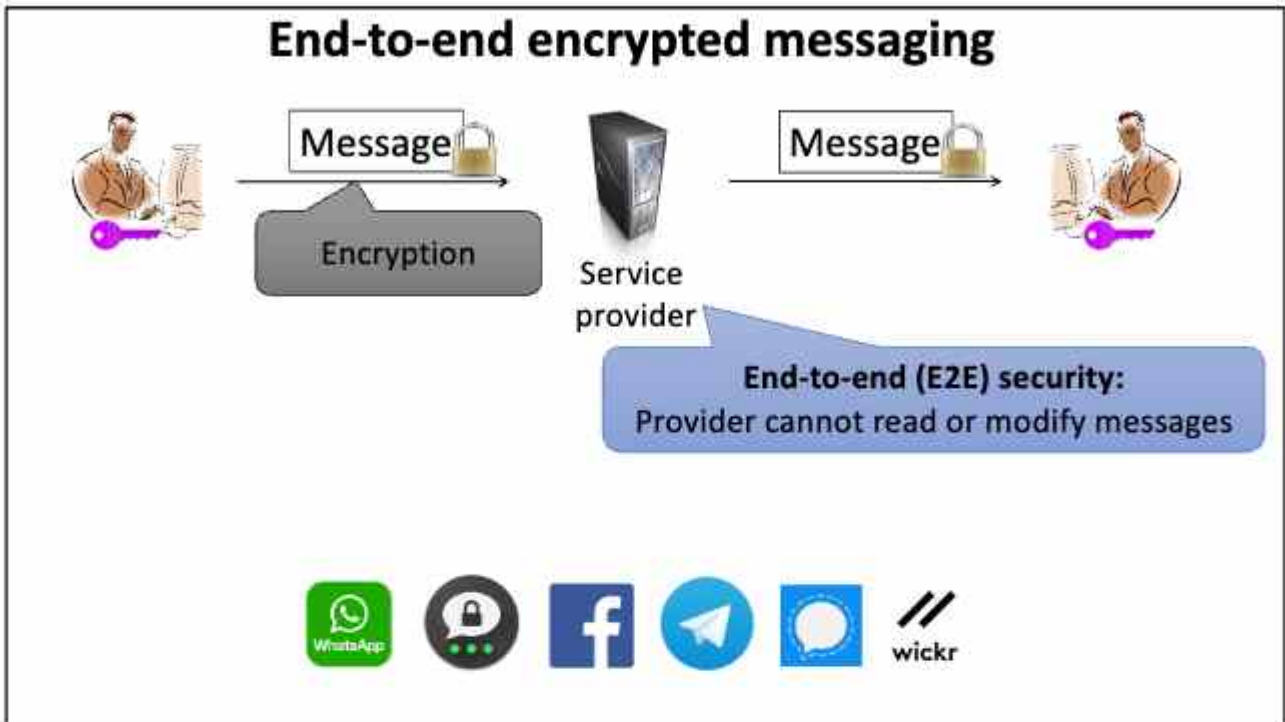
30

Talk Outline

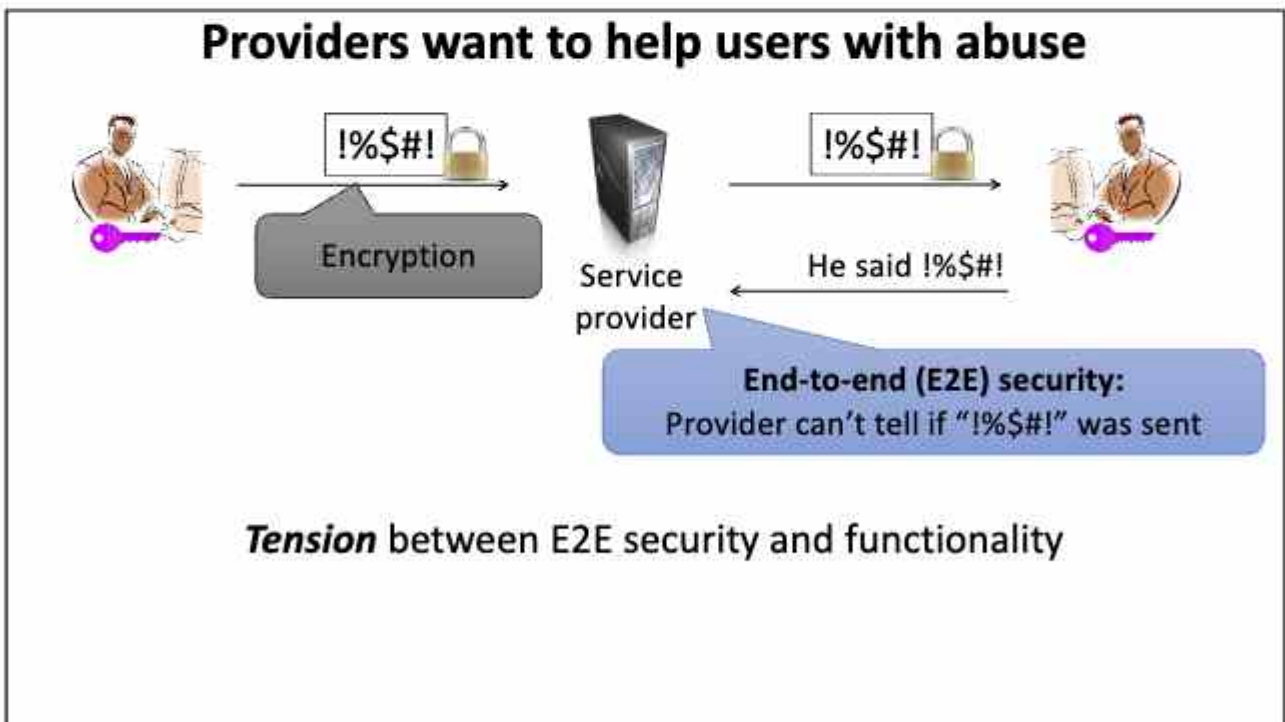
Three case studies:

1. Weak Fiat-Shamir Attacks on Modern Proof Systems
Dao, Miller, Wright, G. (IEEE S&P '23)
2. Attack on Facebook’s Message Franking protocol
Dodis, G., Ristenpart, Woodage (CRYPTO '18)
3. Why Your Encrypted Database Is Not Secure
G., Ristenpart, Shmatikov (HotOS '17)

31



32



33

Background on Message Franking



[Facebook 2016]:

Moderation for E2EE Secret Conversations via cryptographic proof of msg contents. Called technique **message franking**

Lots of academic work on E2EE chat; less on E2EE + moderation. Many questions arise...



- [Frosch et al. 2014]
- [Cohn-Gordon et al. 2016]
- [Bellare et al. 2017]
- [Jaeger and Stepanovs 2018]
- [Coretti et al. 2019]

34

E2EE content moderation



If “proof” verifies, take action against sender (block/ban)

“Proof” !%\$#! was received

35

E2EE content moderation: threat model exercise

1. Who are the entities?
2. What guarantees might the protocol want to make?
3. If some entity is adversarial, how could they try to break guarantees?

Take five minutes to discuss these questions for E2EE content moderation with a partner, then we'll discuss together!

36

Security for E2EE content moderation



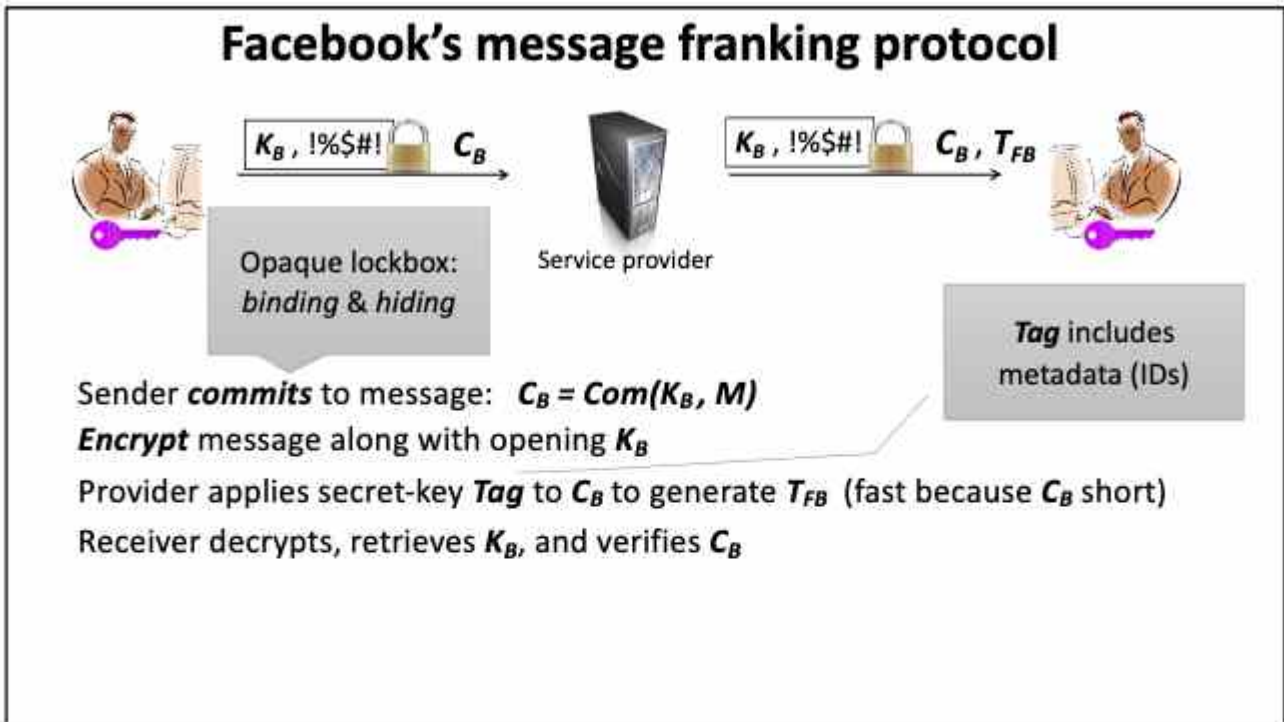
If proof verifies, take action against sender (block/ban)

$0/1 = \text{Verify}(!\%\$\#\!, \pi)$

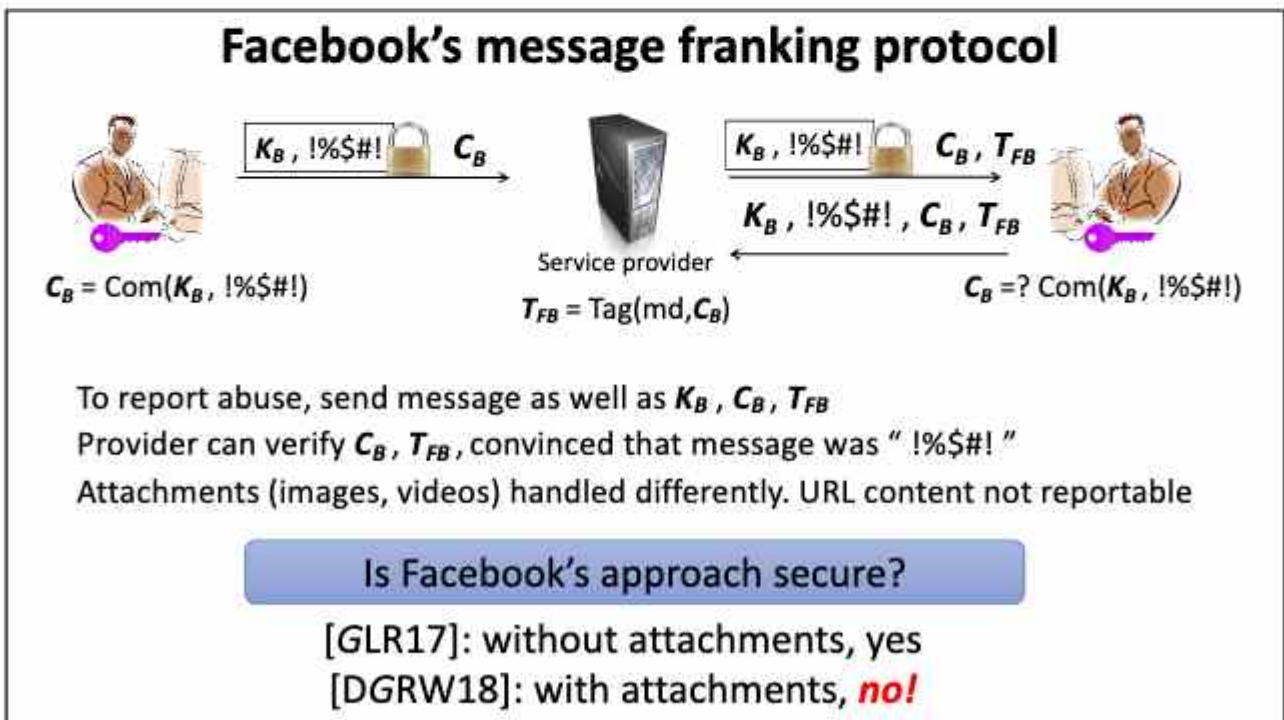
Proof !%\$#! was received

- 1) Receiver can't report a message not sent
- 2) Sender can't send a message that can't be reported
- 3) Confidentiality for unreported messages
- 4) Only service provider can verify reports

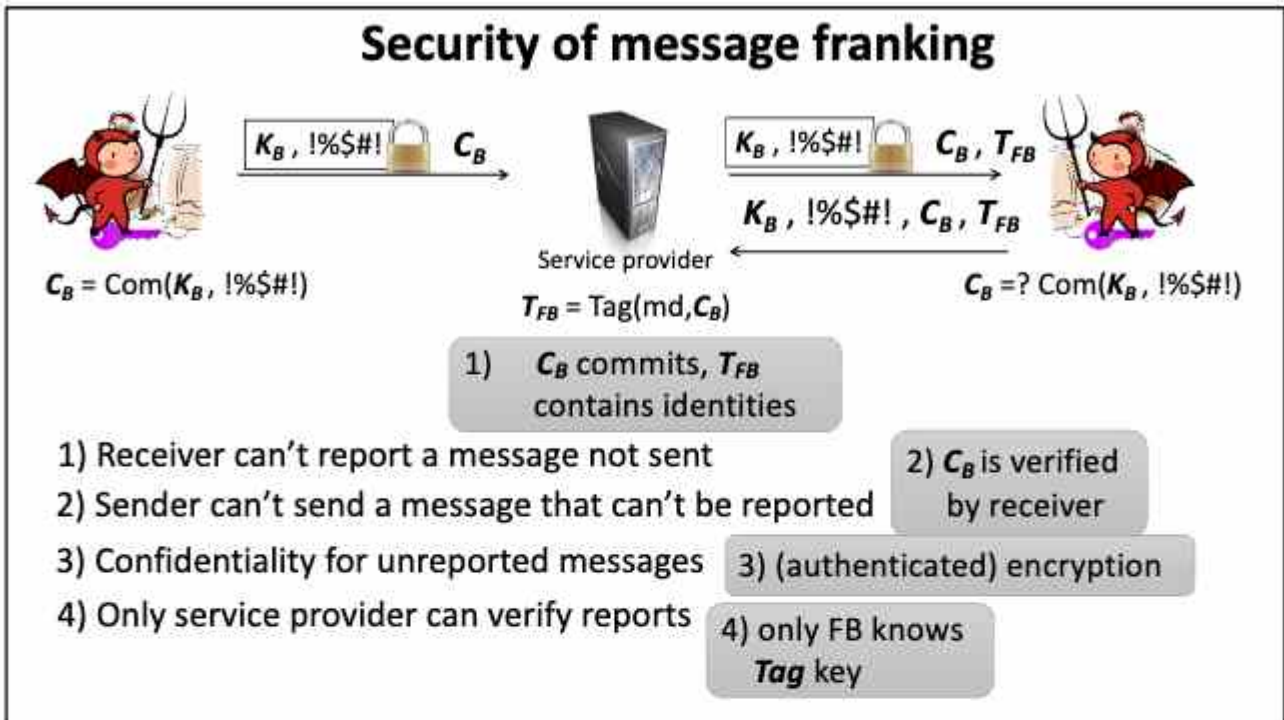
37



38



39



40



41

Facebook's *attachment* franking protocol



Sender **commits** to attachment key: $C_B = Com(K_B, K_{file})$

Encrypt file encryption key K_{file} along with K_B

AES-GCM encrypt attachment: $AES-GCM(K_{file}, file)$

Receiver decrypts as before to get K_{file} and then decrypts attachment

42

Facebook's *attachment* franking protocol

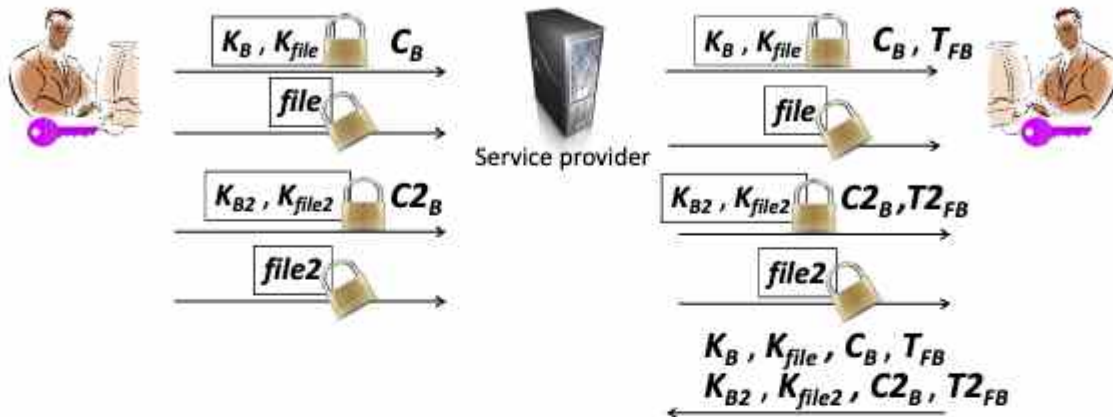


To report abuse, receiver opens K_{file} and other recent messages

Facebook checks openings & decrypts all **unique AES-GCM ciphertexts** to add them to abuse report

43

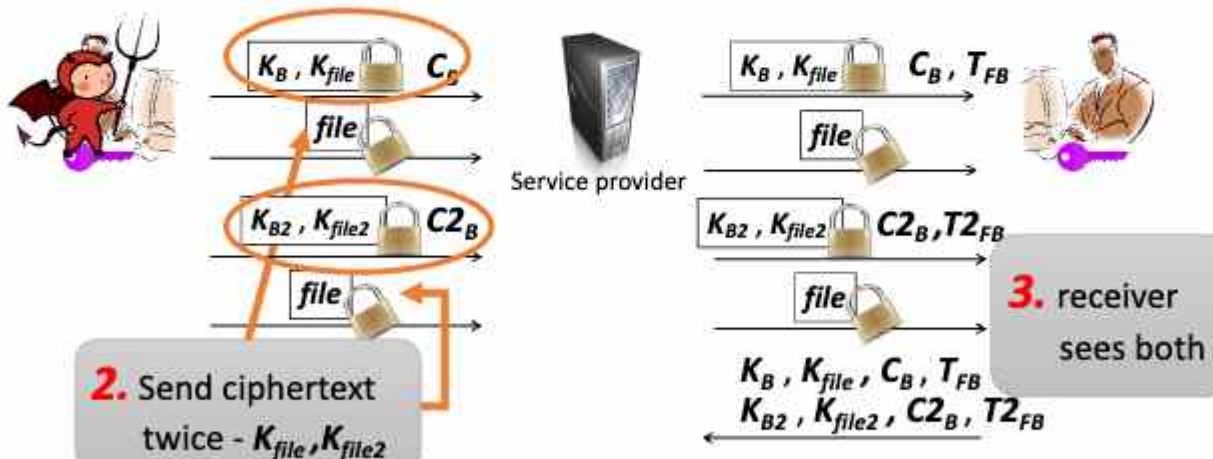
Facebook's attachment franking protocol



To report abuse, receiver opens K_{file} and other recent messages
 Facebook checks openings & decrypts all **unique AES-GCM ciphertexts** to add them to abuse report

44

Our attack exploits AES-GCM



1. Craft special **AES-GCM** ciphertext:

- Decrypts under K_{file} to innocuous image
- Decrypts under K_{file2} to abuse image

4. Only the innocuous image appears in report to Facebook!

45

Our attack exploits AES-GCM



How do we do this?

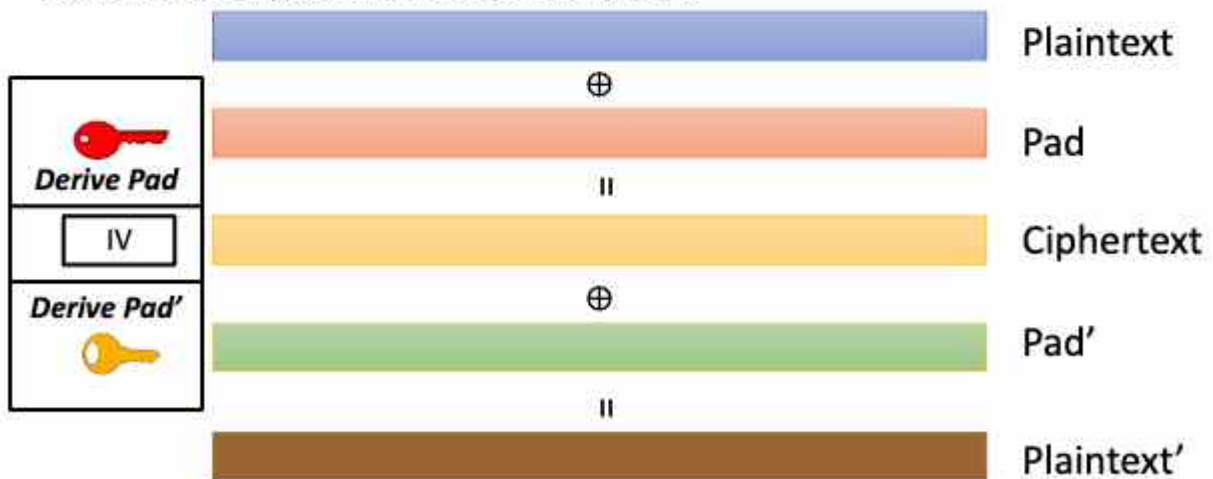


- 1.** Craft special **AES-GCM** ciphertext:
 - Decrypts under K_{file} to innocuous image
 - Decrypts under K_{file2} to abuse image

46

Our attack exploits AES-GCM

Take just encryption part of AES-GCM: CTR mode
 Ciphertext decrypts under two different keys!



47

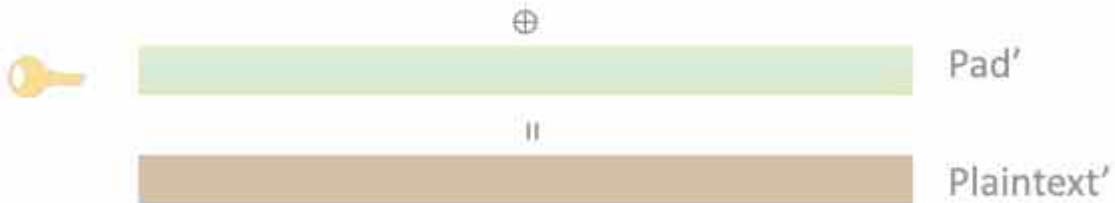
Our attack exploits AES-GCM

Take just encryption part of AES-GCM: CTR mode
Ciphertext decrypts under two different keys!



Two questions remain:

- Won't AES-GCM's MAC prevent decrypting with the wrong key? **No.**
- How do we get both decryptions to be valid images? **It's complicated.**



48

Our attack exploits AES-GCM



1. Craft special **AES-GCM** ciphertext:

- Decrypts under K_{file} to innocuous image
- Decrypts under K_{file2} to abuse image

But isn't **AES-GCM** a secure authenticated encryption scheme?

Yes, but ... this type of attack is not standard
attacker gets to choose K_{file} and K_{file2}

GCM's ciphertexts do not **commit** to plaintexts.

49

Abusive JPEG seen by receiver,
but not in abuse report



Innocuous BMP
in abuse report



Disclosed to Facebook

Thanks to Jon Millican for answering questions!

They fixed by changing report generation logic

Awarded us a bug bounty



50

Key Commitment

- Since this work, several more practical attacks discovered
- Common AEADs do not provide the right guarantees for many applications
- Ongoing work on building key-committing AEAD

Practical Challenges with AES-GCM and the need for a new cipher

Panos Kampanakis
kpanos@amazon.com

Matt Campagna
campagna@amazon.com

Eric Crockett
ericcro@amazon.com

Adam Fetcher
apetcher@amazon.com

Efficient Schemes for
Committing Authenticated Encryption

MERIN BELLARE¹ VIRI TUNG HOANG²

Context Discovery and Commitment Attacks^{*}
How to Break CCM, EAX, SIV, and More

Sanketh Mendu
Cornell Tech

Julia Len
Cornell Tech

Paul Grubbs
University of Michigan

Thomas Ristenpart
Cornell Tech

Partitioning Oracle Attacks

Julia Len Paul Grubbs Thomas Ristenpart
Cornell Tech

How to Abuse and Fix Authenticated Encryption Without Key Commitment

Angelo Albertini¹, Thai Duong¹, Shay Gueron^{2,3}, Stefan Kölbl¹, Atul Luykx¹, and Sophie Schmieg¹

¹Security Engineering Research, Google

²University of Hulls

³Amazon

51

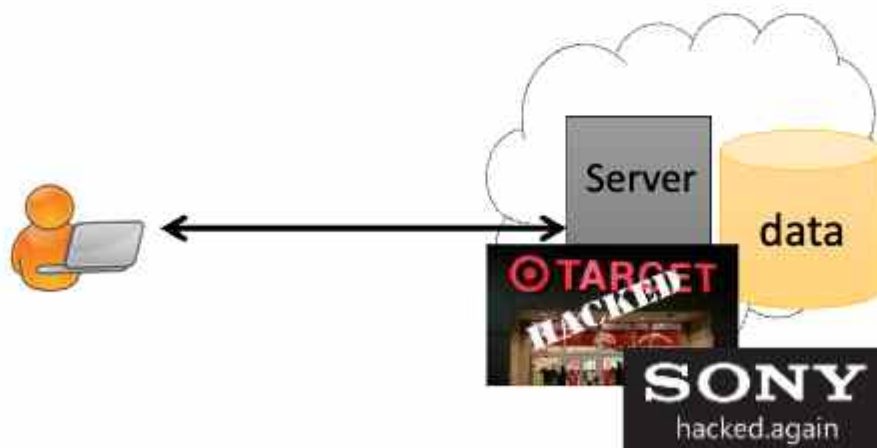
Talk Outline

Three case studies:

1. Weak Fiat-Shamir Attacks on Modern Proof Systems
Dao, Miller, Wright, G. (IEEE S&P '23)
2. Attack on Facebook's Message Framing protocol
Dodis, G., Ristenpart, Woodage (CRYPTO '18)
3. Why Your Encrypted Database Is Not Secure
G., Ristenpart, Shmatikov (HotOS '17)

52

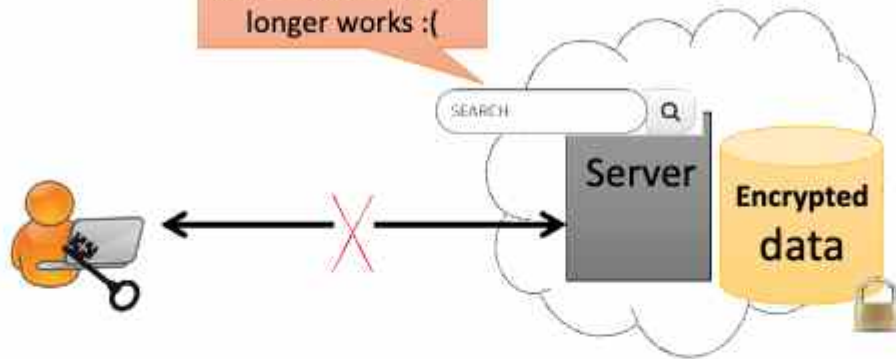
Outsourced applications today



53

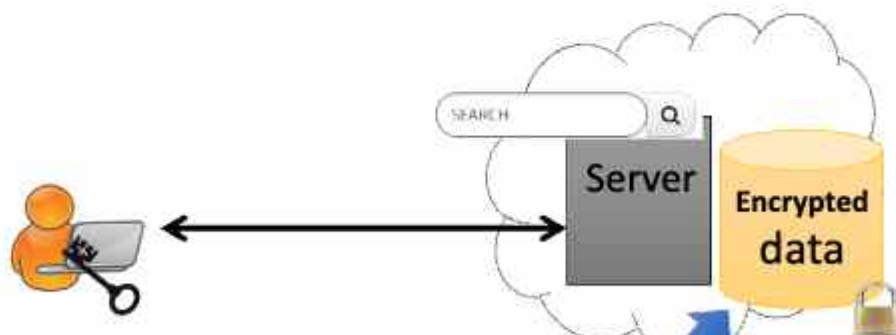
Solution: encrypt the data!

App functionality no longer works :(



54

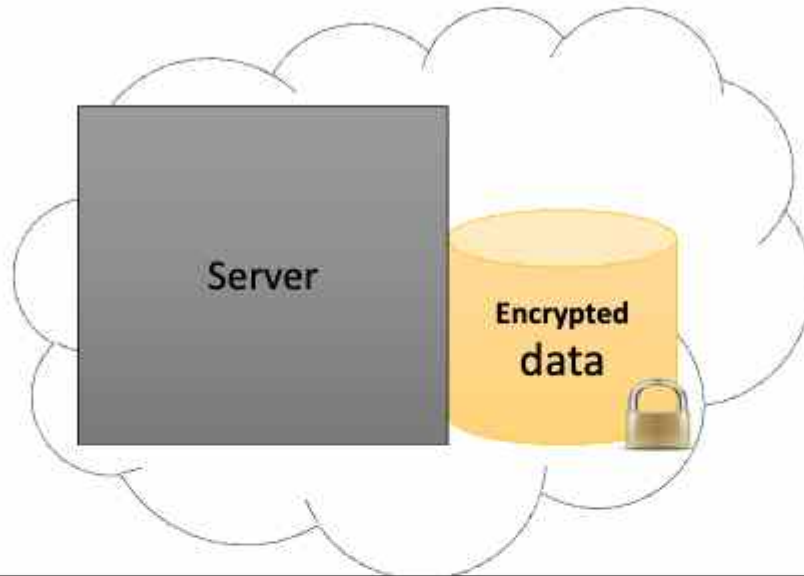
Solution: encrypt the data!



- Searchable Encryption
 - Deterministic Encryption
 - Order-revealing Encryption
- use **property-revealing encryption (PRE)**

55

Building secure systems



56

Building secure systems



57

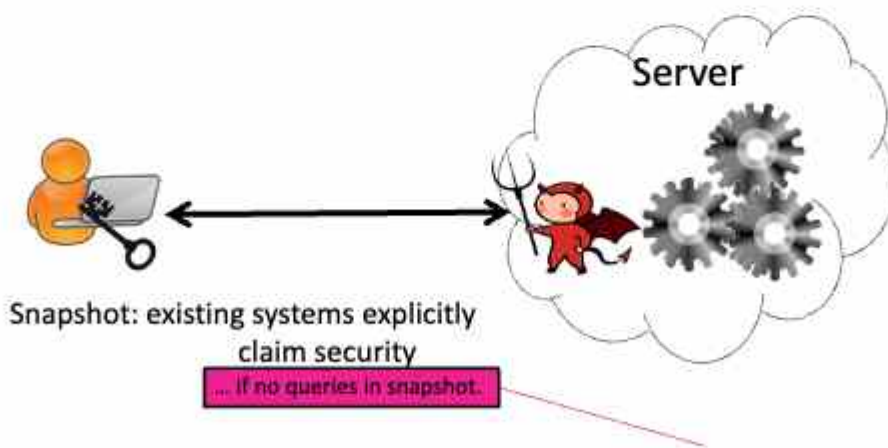
Building secure systems

- CryptDB (SOSP '11)
- Mylar (NSDI '14)
- Seabed (OSDI '16)
- Arx (VLDB '19)
- Others
- Lots of industry/government interest!!



58

Threat models



60

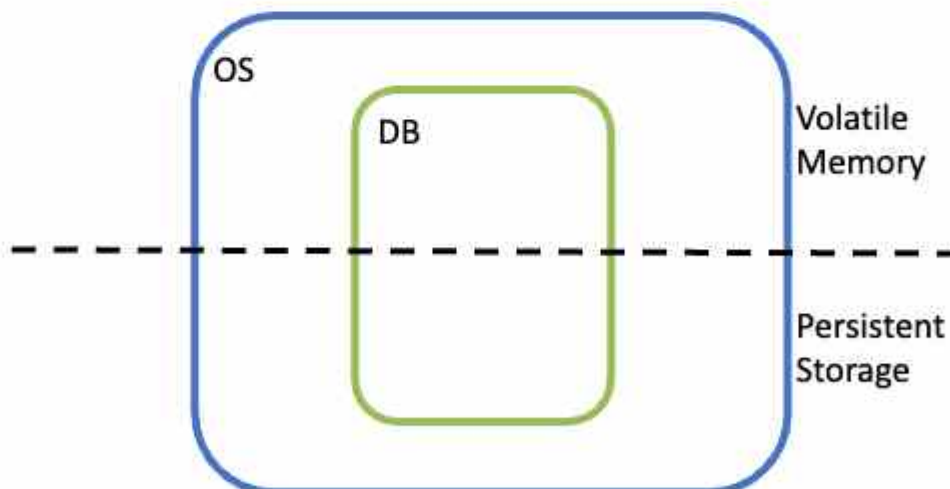
Snapshot: existing papers explicitly claim security
if no queries are in the snapshot

This is false. Real snapshots have query information.

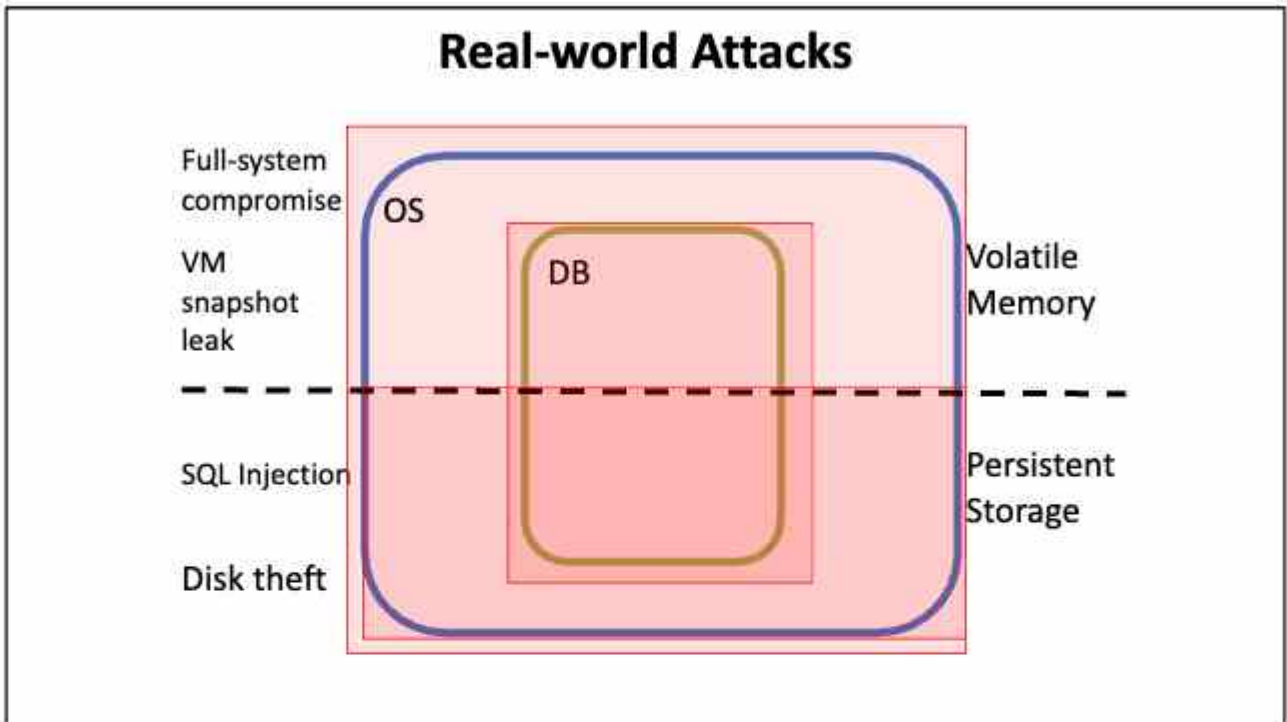
Implication is confidentiality loss in
Seabed, CryptDB, Mylar, Lewi-Wu, etc.

61

System Abstraction




62



63

Case Study: MySQL

MySQL is running example, but other widely-used DBMS's have these features



The MySQL logo is a stack of three silver cylinders with the word 'MySQL' in blue and yellow text below them.

64

Threat model	What MySQL leaks	Attack enabled against
Disk theft	MVCC data structures	Arx's range query index
SQL Injection	Past query statistics	Seabed's SPLASHE scheme
Full system compromise or VM snapshot leak	Text of past queries	CryptDB, Lewi/Wu, etc.

65

Disk theft

Healthcare IT News

Privacy & Security

Stolen laptop leads to breach notification for 20,000 Lifespan patients



SC Magazine US > Blog > The Data Breach Blog > Hard drive stolen from Jackson Memorial Hospital

Hard drive stolen from Jackson Memorial Hospital

What happened? The hard drive was stolen from the hospital's data center, which is secured by cyberlocks and swipe cards. Several dozen people have access to the center,

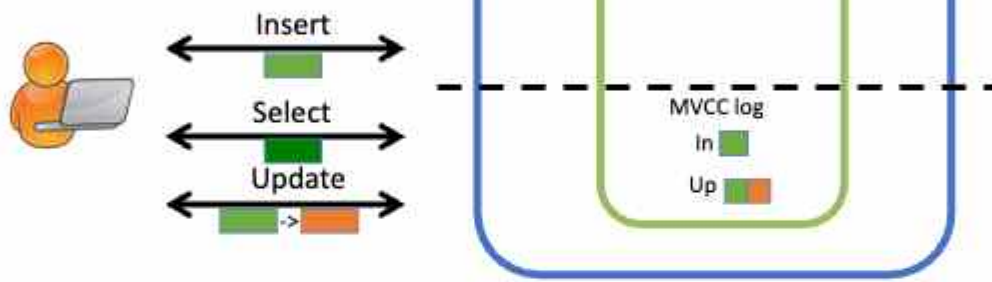
66

Logs on disk

General query log records all queries, but is not widely used.
Binary log records modifications, used for replication and recovery.

Data modification queries can be reconstructed from these logs
 [FHMW '10,FKSHW '12]

Modern SQL databases achieve **multi-version concurrency control** using log data structures

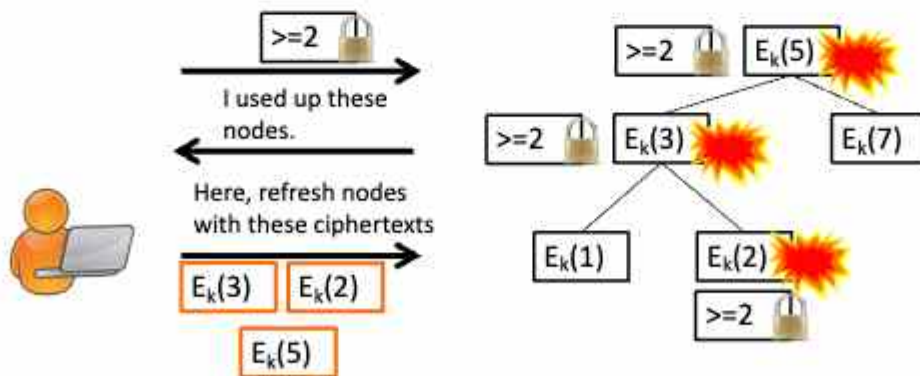


67

Arx (VLDB '19)

Poddar et al.

Range queries via chained garbled circuits: tree nodes become consumed, need replacing



68

Logs on disk + Arx *Poddar et al.*

Range queries via chained garbled circuits: tree nodes become consumed, need replacing

Consumed nodes immediately replaced – stored in MVCC log!

Query access pattern is recorded on disk. Can recover queries and plaintexts using [GSBNR] or [LMP]

Here, refresh nodes with these ciphertexts

$E_k(3)$

$E_k(2)$

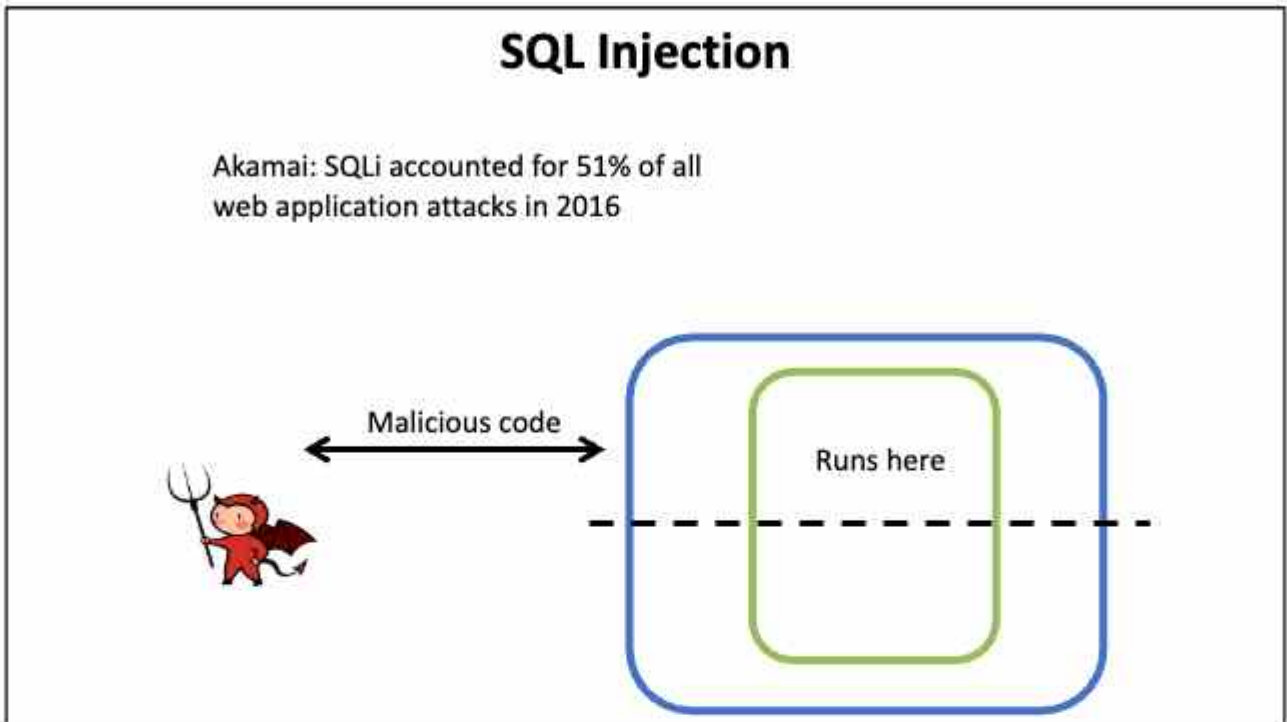
→

$E_k(5)$

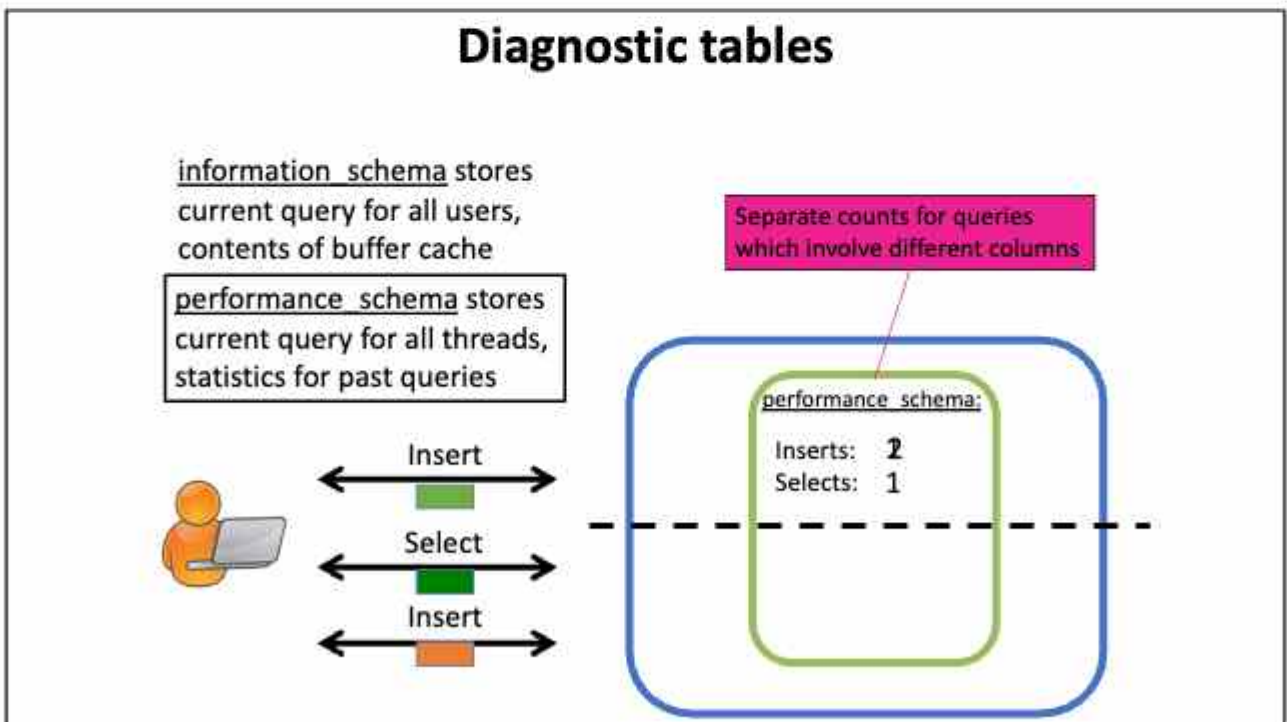
69

Threat model	What MySQL leaks	Attack enabled against
Disk theft	MVCC data structures	Arx's range query index
SQL Injection	Past query statistics	Seabed's SPLASHE scheme
Full system compromise or VM snapshot leak	Text of past queries	CryptDB, Lewi/Wu, etc.

70

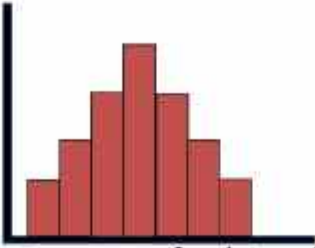


71



72

Seabed (OSDI '16)



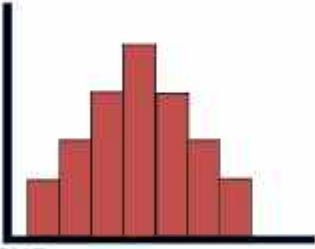
Name	Has malaria
Paul Grubbs	1
Thomas Ristenpart	0
Vitaly Shmatikov	0

Plaintext recovery for deterministic encryption in CryptDB by **frequency analysis**: match data histogram to auxiliary model [NKW]

"Big Data Analytics over Encrypted Datasets with Seabed"
Papadimitriou et al.

73

Seabed (OSDI '16)



Name	Has malaria
Paul Grubbs	1
Thomas Ristenpart	0
Vitaly Shmatikov	0

↓ ("Has malaria"=1) ("Has malaria"=0)

Name	C2	C3
aspoiwnpoinio	$E_k(1)$	$E_k(0)$
petryoieuytiew	$E_k(0)$	$E_k(1)$
Xncmxncmbcn	$E_k(0)$	$E_k(1)$

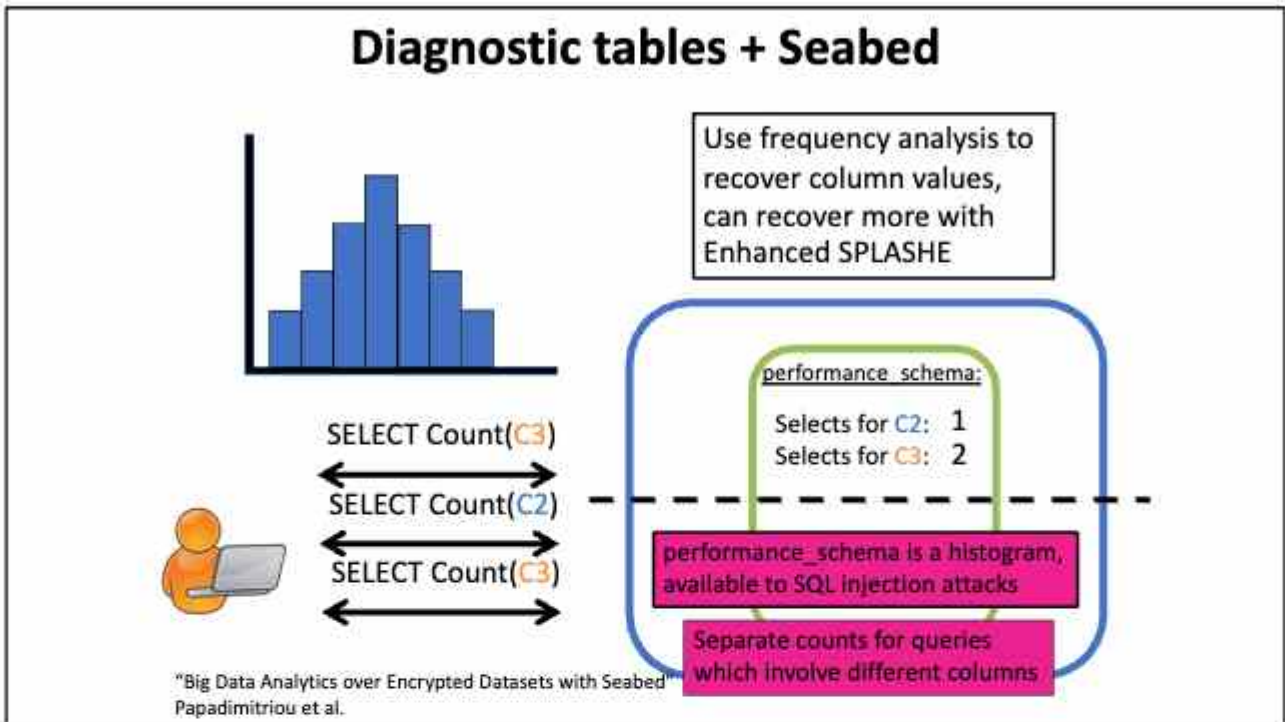
SELECT Count("Has malaria") WHERE "Has malaria"=1 → SELECT Count'(C2)

Separate counts for queries which involve different columns

SPLASHE:
Each possible plaintext gets its own column.
WHERE clause transformed to correct column.

"Big Data Analytics over Encrypted Datasets with Seabed"
Papadimitriou et al.

74



75

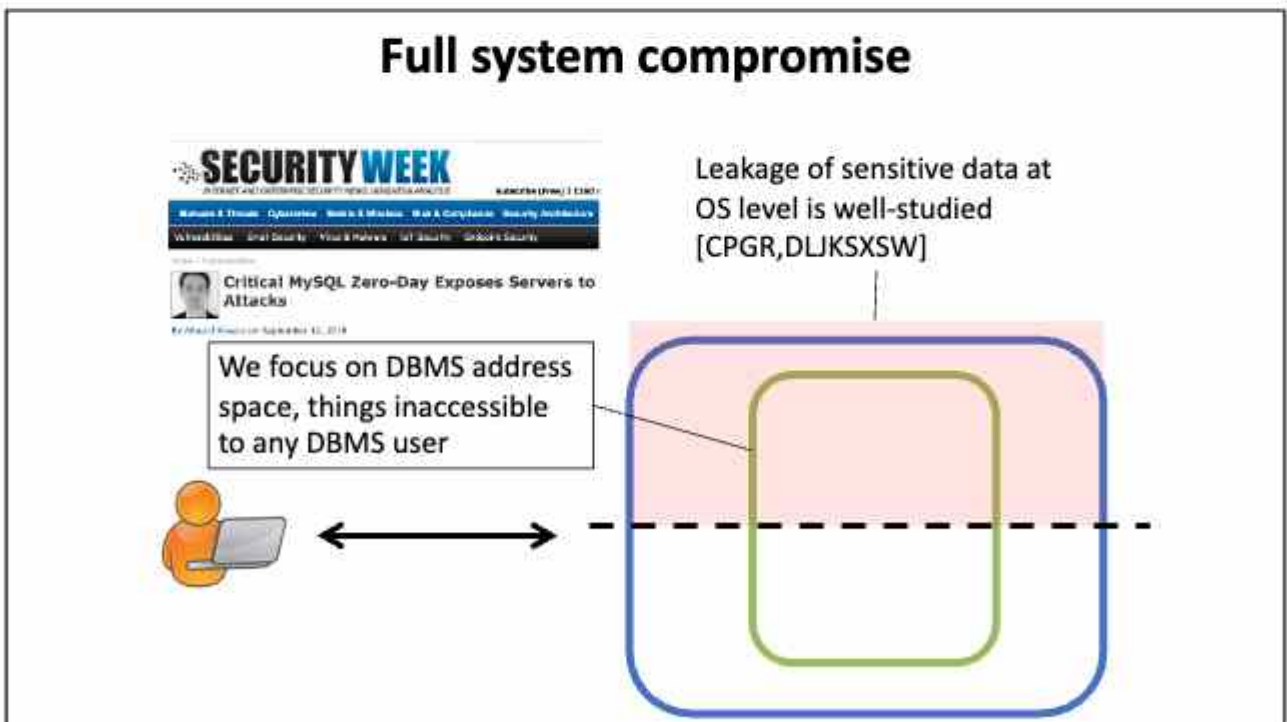
Diagnostic tables + Seabed

Plaintext Schema		Schema with Enhanced SPLASHE			
country	salary	country	salaryUSA	salaryCanada	salaryOthers
USA	100000	DET(Chile)	ASHE(100000)	ASHE(0)	ASHE(0)
USA	100000	DET(Iraq)	ASHE(100000)	ASHE(0)	ASHE(0)
Canada	200000	DET(China)	ASHE(0)	ASHE(200000)	ASHE(0)
USA	300000	DET(Japan)	ASHE(300000)	ASHE(0)	ASHE(0)
Canada	500000	DET(Israel)	ASHE(0)	ASHE(500000)	ASHE(0)
Canada	800000	DET(U.K.)	ASHE(0)	ASHE(800000)	ASHE(0)
India	100000	DET(India)	ASHE(0)	ASHE(0)	ASHE(100000)
India	100000	DET(India)	ASHE(0)	ASHE(0)	ASHE(100000)
Chile	200000	DET(Chile)	ASHE(0)	ASHE(0)	ASHE(200000)
Iraq	300000	DET(Iraq)	ASHE(0)	ASHE(0)	ASHE(300000)
China	500000	DET(China)	ASHE(0)	ASHE(0)	ASHE(500000)
Japan	800000	DET(Japan)	ASHE(0)	ASHE(0)	ASHE(800000)
Israel	130000	DET(Israel)	ASHE(0)	ASHE(0)	ASHE(130000)
U.K.	210000	DET(U.K.)	ASHE(0)	ASHE(0)	ASHE(210000)

76

Threat model	What MySQL leaks	Attack enabled against
Disk theft	MVCC data structures	Arx's range query index
SQL Injection	Past query statistics	Seabed's SPLASHE scheme
Full system compromise or VM snapshot leak	Text of past queries	CryptDB, Lewi/Wu, etc.

77

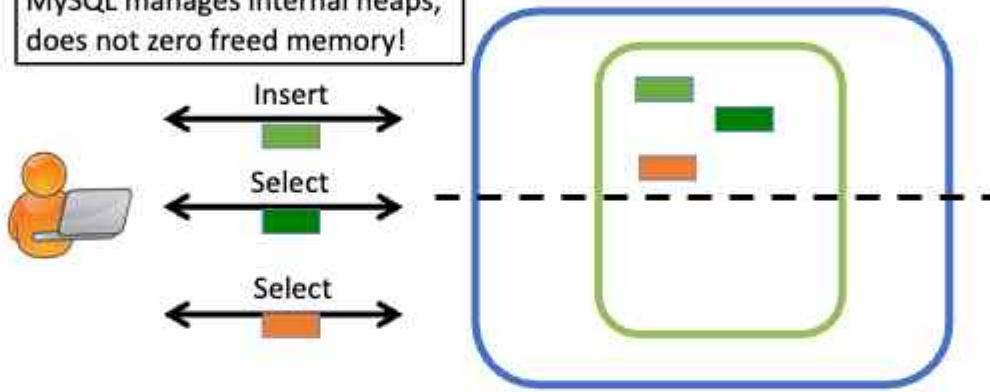


78

Data structures and caches

“Adaptive hash index” tracks accesses to pages, creates index over page automatically.
 MySQL query cache stores select queries and results, other query caches (memcached).

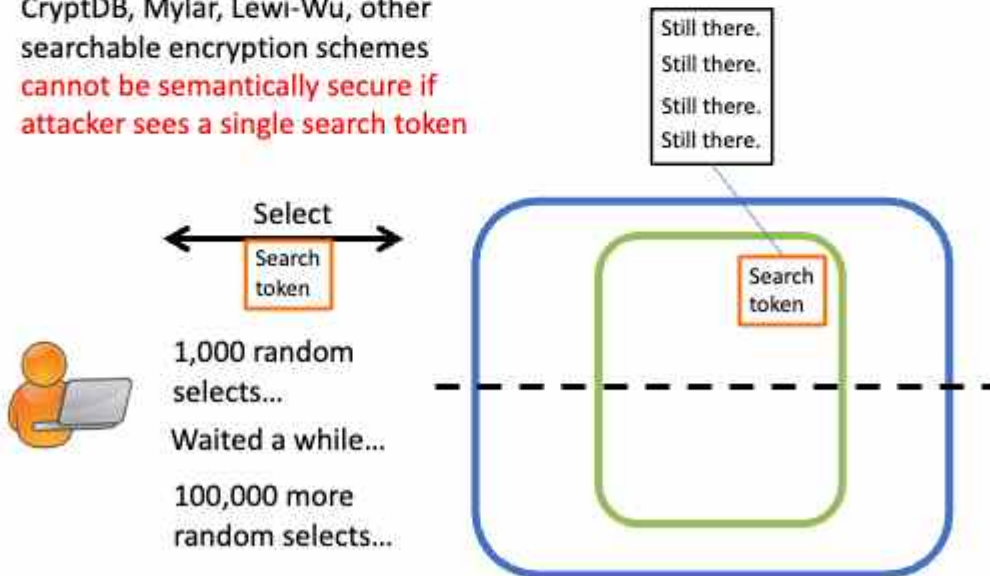
MySQL manages internal heaps, does not zero freed memory!



79

Token-based systems

CryptDB, Mylar, Lewi-Wu, other searchable encryption schemes **cannot be semantically secure if attacker sees a single search token**



80

Snapshot attacks

Recent work on MongoDB's queryable encryption showed attacks like these against a real system



Security Analysis of MongoDB Queryable Encryption

Zichen Gui, Kenneth G. Paterson, and Tianxin Tang
Department of Computer Science, ETH Zurich, Zurich, Switzerland

81

Summary of case studies

82

Takeaways

- Attacks are fun!
- To find new attacks, start with threat models.
- Understanding security proofs is hugely helpful
- Look to new kinds of cryptography being deployed
 - ZKP, MPC, PIR, encrypted search, ...
- Interplay between cryptography and systems is complex, subtle

Thanks for listening!
Any questions?

paulgrub@umich.edu

@pag_crypto

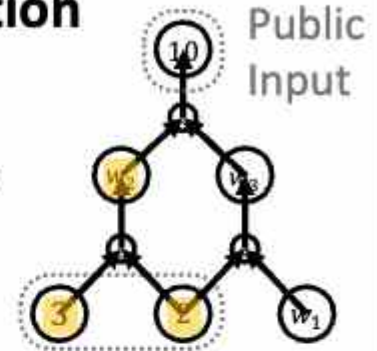
83

84

Plonk - Protocol Description

Constraint System:

- Gate Vectors: $\vec{a} = (3, 2, w_2)$, $\vec{b} = (2, w_1, w_3)$, $\vec{c} = (w_2, w_3, 10)$
- Gate Constraints: $a_1 + b_1 = c_1$, $a_2 \times b_2 = c_2$, $a_3 \times b_3 = c_3$
- Consistency Constraints: $a_2 = b_1$, $a_3 = c_1$, $b_3 = c_2$



Verification Equation:

$$\sum_{i=1}^k \text{PI}_i \cdot L_i(\zeta) + \underbrace{\text{PI}(\zeta) + \text{Eq}(\zeta)}_{\text{Gate Check}} + \underbrace{\alpha \cdot \text{Per}(\zeta) + \alpha^2 \cdot (z(\zeta) - 1)L_1(\zeta)}_{\text{Consistency Check}} = \underbrace{Z_H(\zeta) \cdot t(\zeta)}_{\text{Vanishing Domain}}$$

Annotations: (Fixed) Scalars, Batching Challenge, Evaluation Point

94

Plonk - Protocol Description

Verification Equation:

$$\sum_{i=1}^k \text{PI}_i \cdot L_i(\zeta) + \underbrace{\text{PI}(\zeta) + \text{Eq}(\zeta)}_{\text{Gate Check}} + \underbrace{\alpha \cdot \text{Per}(\zeta) + \alpha^2 \cdot (z(\zeta) - 1)L_1(\zeta)}_{\text{Consistency Check}} = \underbrace{Z_H(\zeta) \cdot t(\zeta)}_{\text{Vanishing Domain}}$$

Annotations: (Fixed) Scalars

95

Plonk - Weak Fiat-Shamir Attack

Verification Equation:

$$\sum_{i=1}^k \underbrace{PI_i}_{\text{(Fixed) Scalars}} \cdot \underbrace{L_i(\zeta)}_{\text{Linear Equation}} + \underbrace{PI(\zeta)}_{\text{Gate Check}} + \underbrace{\alpha \cdot Per(\zeta) + \alpha^2 \cdot (z(\zeta) - 1)L_1(\zeta)}_{\text{Consistency Check}} = \underbrace{Z_H(\zeta) \cdot t(\zeta)}_{\text{Vanishing Domain}}$$

Weak F-S Attack: When PI is not part of hash computation (for deriving α, ζ)

1. Select arbitrary polynomials for the proof \Rightarrow compute all evaluations except $PI(\zeta)$.
2. Solve for the public values $PI = (PI_1, \dots, PI_k)$ that will pass verification.

Degrees of freedom: can set all but one PI_i to be arbitrary.

In Contrast: For strong Fiat-Shamir, changing PI will also change α, ζ .