black hat ASIA 2014

Say it Ain't So

An Implementation of Deniable Encryption

Ari Trachtenberg

Electrical and Computer Engineering
Boston University
http://nislab.bu.edu   *trachten@bu.edu*

BOSTON UNIVERSITY

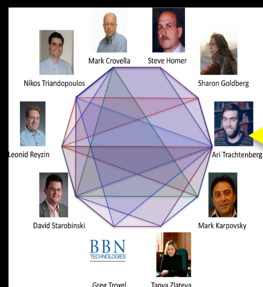
was:
Visiting Professor at
Technion – Israel Institute of Technology

TCE Technion Computer Engineering

Why listen ... to me?

Broader effort on phone security: http://www.bu.edu/riscs/

BU Center for Reliable Information Systems & Cyber Security



me

Supported by NSF:

This material is based upon work supported by the National Science Foundation under Grant No. ~~(grantee must enter NSF grant number)~~.

CNS-1012910

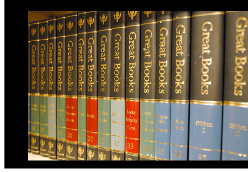
Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

I'm the only one talking?

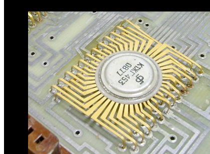
Contents

Main idea

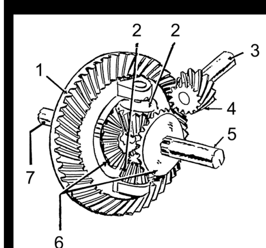
What's out there



Core Tech.



Implementation

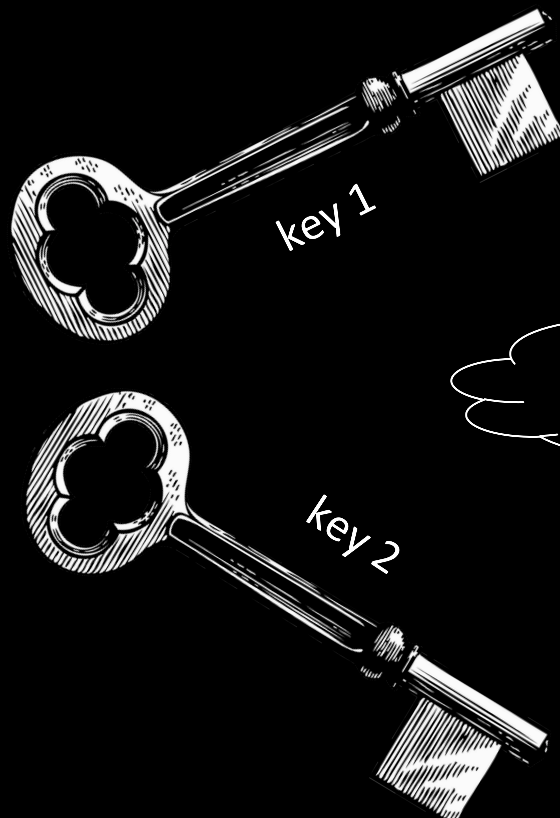


Conclusions

Main Idea

Encryption

הַיְרִיָה את מה אז נח
נִפְגְעָה רה בַבְּצָה .
בוא עלם אני רעד



To make coke mix caffeine with citric acid and water.

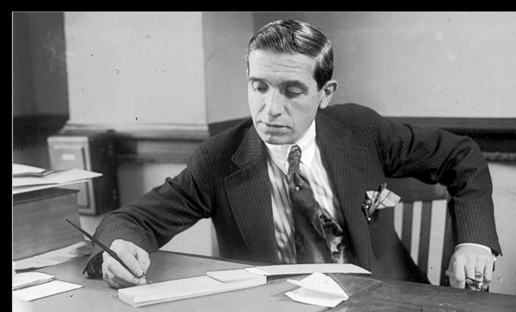
plausible deniability

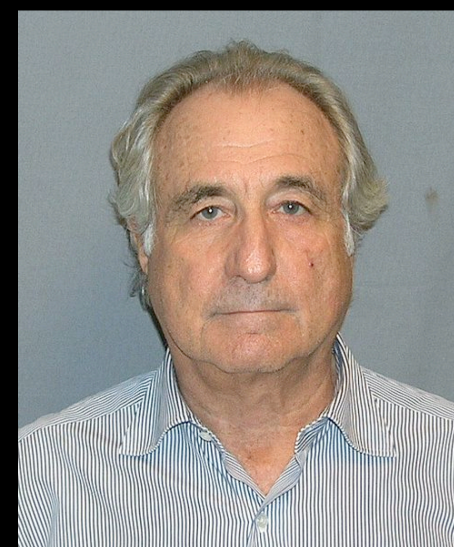
To make coke mix coal with heat and water.

Main Idea

But wait …

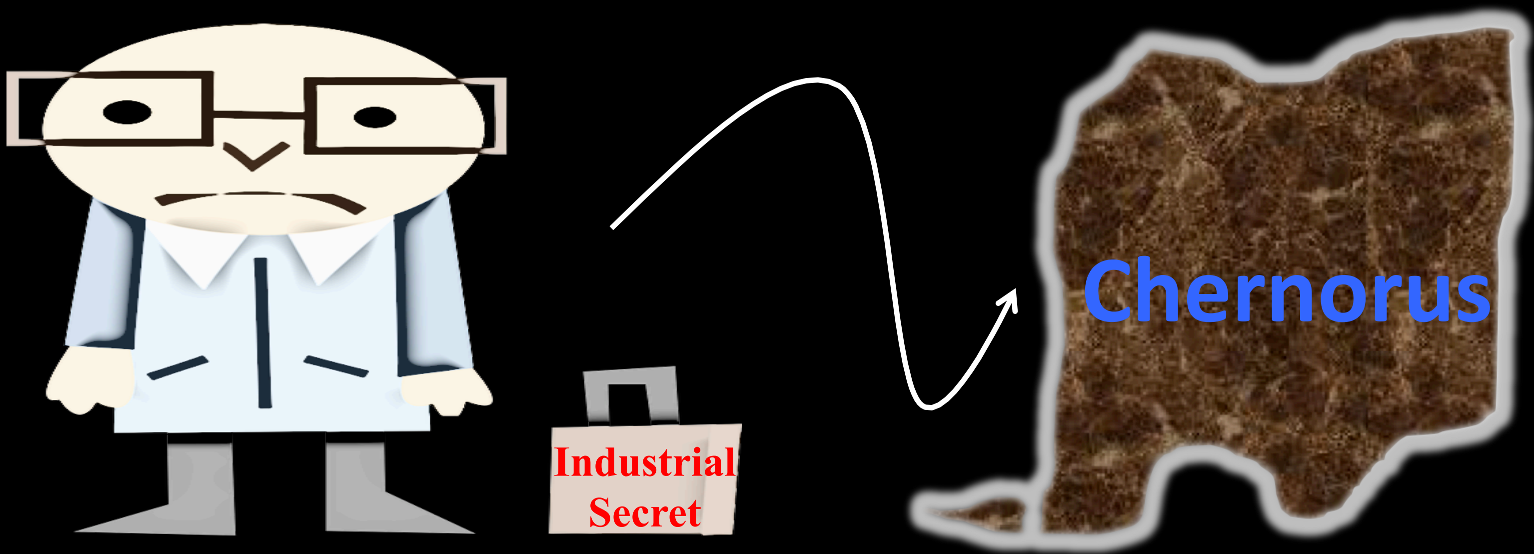
deniability $=^?$ lying $=^?$ bad





US Dept. of Justice

Motivation



Plaintext:

To make coke mix caffeine with citric acid and juice.

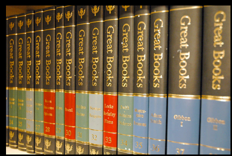
→

Encryption:

הַיְרִיָּה את מה אז נח
נִפְגְעָה רה בַּבִּצָה.
בוא עלם אני רעד

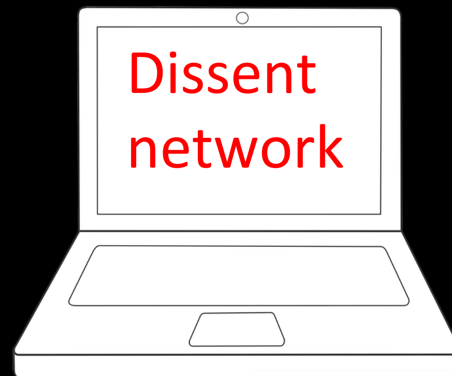
Motivation

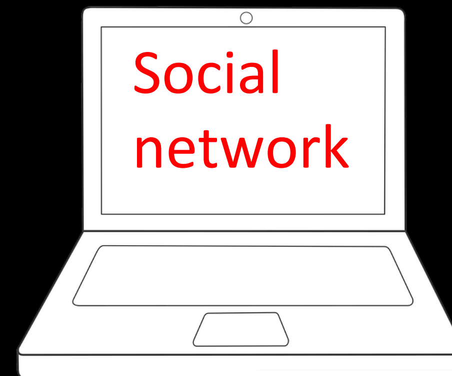
To make coke mix caffeine with citric acid and water.



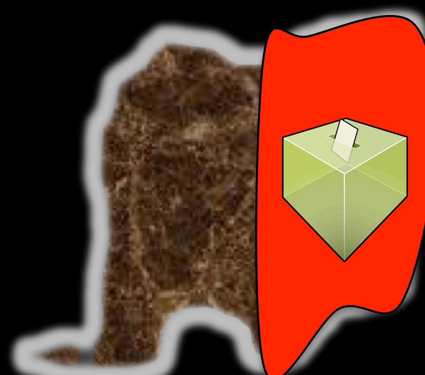
Other applications

- Secure storage

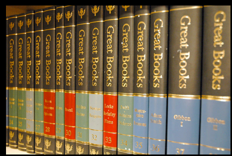




- Electronic voting

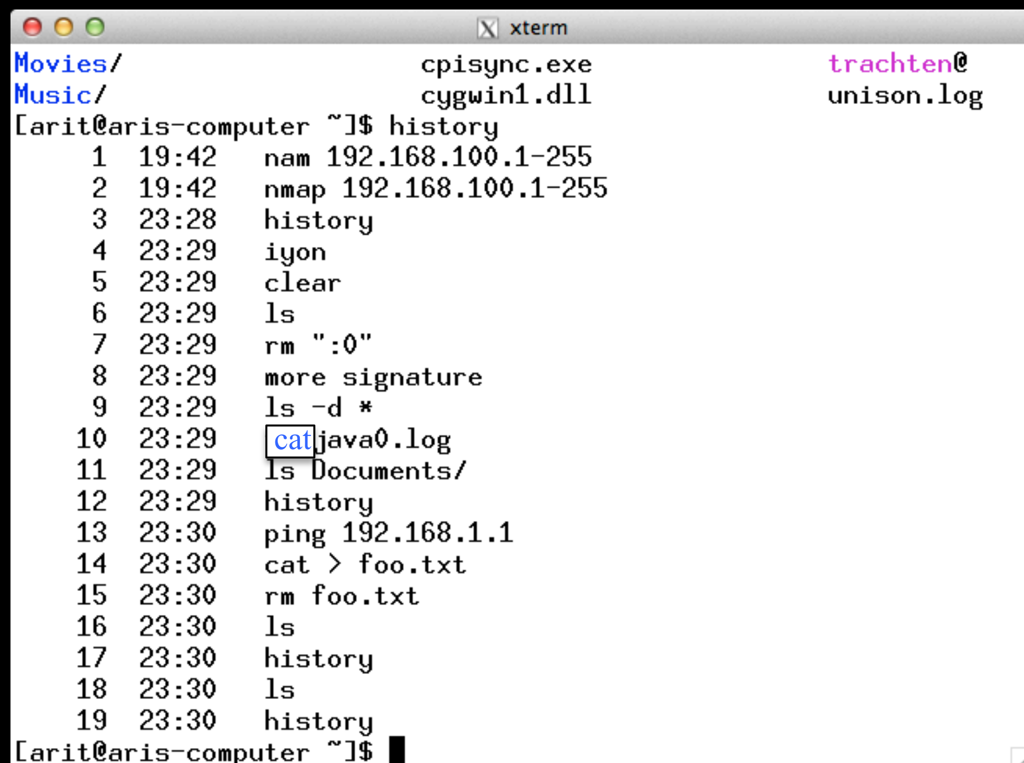


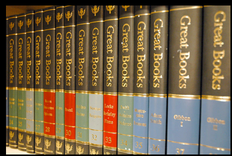
- Censorship



??? applications

- Deniable logs
  - sysadmin
  - accounting





Our Solution

Plaintext:

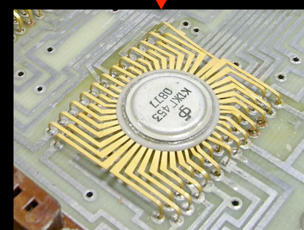
To make coke mix caffeine with citric acid and water.

Decoy 1:

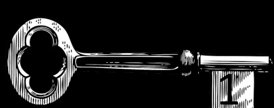
To make coke mix coal with heat and water.

Decoy 2:

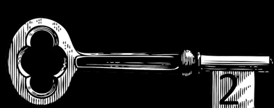
To make smoke mix coal with acid and water.



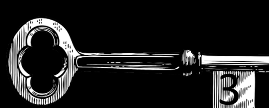
הַיְרִיָּה את מה אז נח נִפְגְּעָה רה
בַּבִּצָּה . בוא עלם אני רעד



To make coke mix caffeine with citric acid and water.

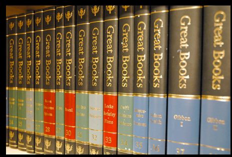


To make coke mix coal with heat and water.



To make smoke mix coal with acid and water.

To make a joke mix your calf in citric acid and water.

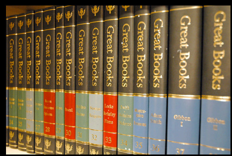


Our Solution

Features:

(n-character plaintext)

- *"Short" encryption*: *~n log n* bits
  - allows incremental modification
- *"Short" key*: *~log n bits*
  - plan-ahead
  - after-the-fact

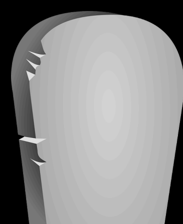


What's out there?

Disk Encryption

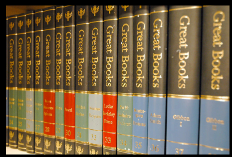
- Rubberhose File System
  Assange, Dreyfus, Weinmann '00

- FreeOTFE



- StegFS
  Mcdonald and Kuhn, based on Andreson, Needham, Shamir '98

- TrueCrypt



Simple solution

One-time PAD

Text: `To make coke mix caffeine with citric acid and water.`

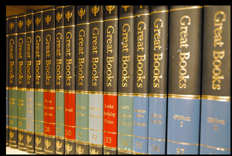
Key: `asdk;asdlfkm2309jaslk2m3-sa-0dsadf92-30asd9vmsamw;qla`

Ciphertext: `uhoiuhonhcvv6876guyk8b8976tgm 90867y n56976t087huni8`

Key2: `asdk;asdlfkm2309jaslk2m3-sa-0dsadf92-30asd9vmsamw;qla`

Text2: `To make coke mix coal with heat and water.`

looooooooong key!



Smarter solutions

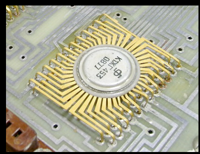
Crypto

- Deniable Encryption
  Canetti, Dwork, Naor, Ostrovsky '97

- Pracitcal Deniable Encryption
  Klonowski, Kubiak, Kutyowski '08

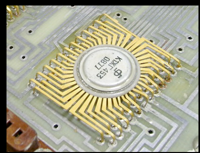
- Bideniable public-key encryption
  O'Neill, Peikert, Waters '11

- …



Core Technology

- Set reconciliation
- String reconciliation
- Unique decoding



Core Tech – *Set Reconciliation*

Alice

Bob

T. Homeless H. →

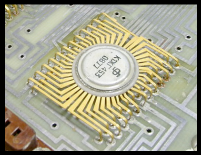
← D.S. Cowboy

Set A (Alice):

K. Burnett
011 7 123-456
D. Dante
(314) 159-2653
T. Homless H.
333-4???
Solo
x2-0000
R.T. Maurice
253-6776
C. Crunch
260-0KHZ

Set B (Bob):

K. Burnett
011 7 123-456
D. Dante
(314) 159-2653
Solo
x2-0000
R.T. Maurice
253-6776
C. Crunch
260-0KHZ
D.S. Cowboy
1-996-KUJI



Core Tech – *Set Reconciliation*

The **BIG** Idea

$S_A = \{x_1, x_2, x_3, \ldots x_n\}$

degree $|S_A|$

S_B

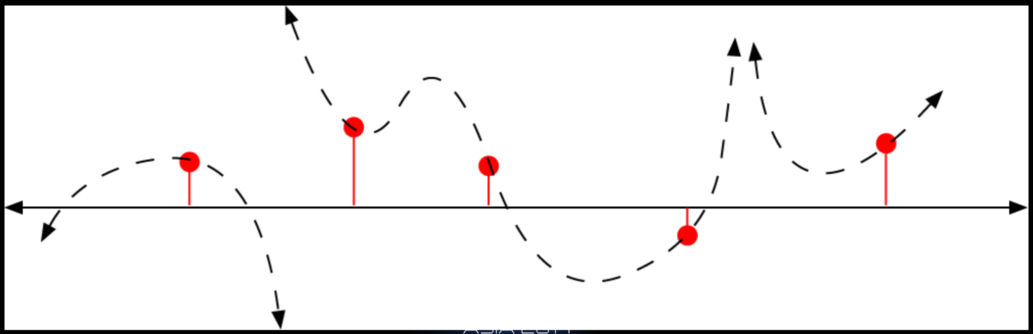
$\chi_{S_A}(Z) = (Z - x_1)(Z - x_2)(Z - x_3)\ldots(Z - x_n)$

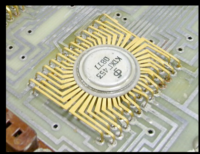
$\chi_{S_B}(Z)$

missing data:

$\chi_{\text{missing data}}(Z) \;=\; \chi_{S_A}(Z) / \chi_{S_B}(Z) \;=\; \chi_{\Delta_A}(Z) / \chi_{\Delta_B}(Z)$

degree $|S_A \oplus S_B|$





Core Tech – *String Reconciliation*

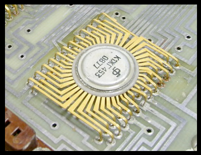
programming language PL/I by Bruce Walker

σ = **IF IF = THEN THEN THEN = ELSE ELSE ELSE = IF;**

- THEN

+ ELSE

τ = **IF IF = THEN THEN = ELSE ELSE ELSE = IF ELSE;**



Core Tech – *String Reconciliation*

σ = IF_IF=THEN_THEN_THEN=ELSE_ELSE_ELSE=IF;

shingle size *l*= 3

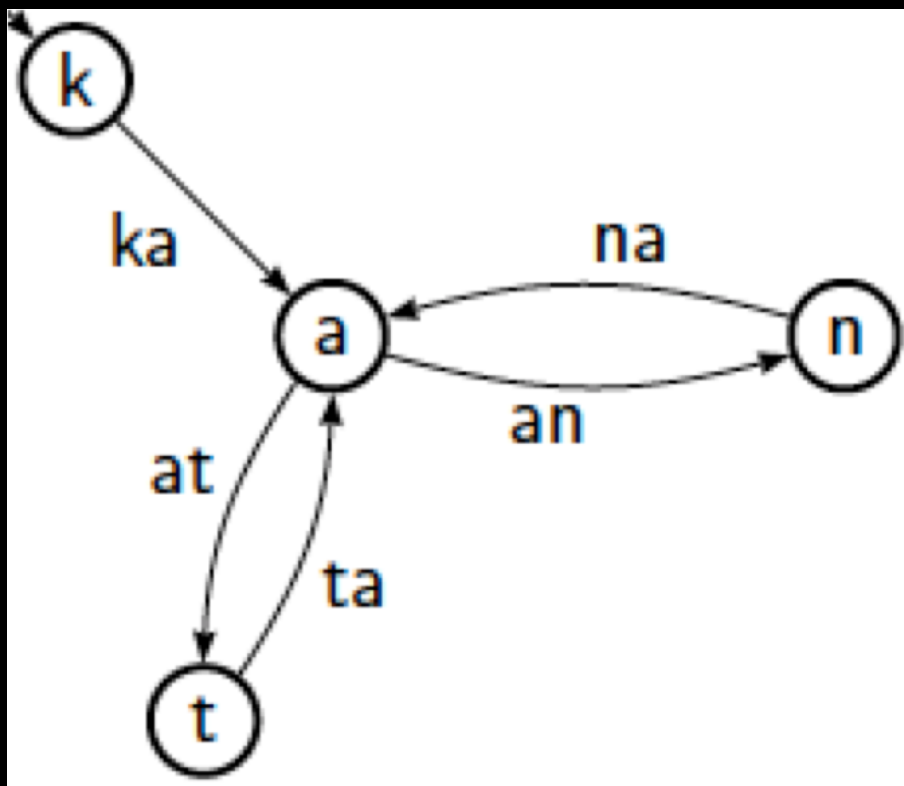
{IF_,F_I, _IF, IF=, F=T, ..., THE, HEN, EN_, N_T, _TH, THE, HEN, EN=,...}

set reconcile

{IF_,F_I, _IF, IF=, F=T, ..., THE, HEN, EN=,...}

τ = IF_IF=THEN_THEN=ELSE_ELSE_ELSE=IF_ELSE;

Core Tech – *Unique Decoding*



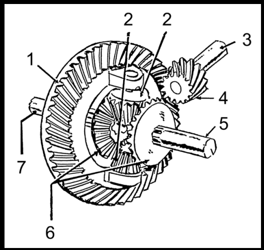
Two decodings:

katana

kanata

Solutions:

- increase shingle length
- merge shingles



Implementation

p= To make coke mix caffeine with citric acid and juice.

Encryption

1. Shingle p
2. Pick some of the shingles
3. Generate the characteristic polynomial
4. Evaluate at several points

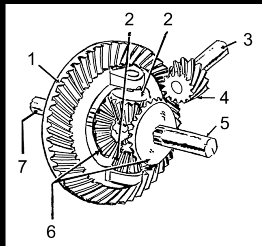
S_p= {To_, o_m, mak, ake, ke_, e_c, _co, cok, oke, ke_, e_m, _mi, mix, ...}

S_i= {To_, o_m, ~~mak~~, ake, ke_, e_c, _co, cok, oke, ke_, e_m, _mi, ~~mix~~, ...}

$\chi_{S_i}(Z)$= (Z-[To])(Z-[o m]) (Z-[ake]) (Z-[ke])(Z-[co]) ...

Ciphertext: $\chi_{S_p}(0)$, $\chi_{S_p}(15)$, $\chi_{S_p}(51)$, $\chi_{S_p}(60)$, $\chi_{S_p}(85)$, $\chi_{S_p}(90)$, $\chi_{S_p}(102)$, $\chi_{S_p}(105)$

Key: [mak, e_c, mix, cok, e_m]



Implementation

Decryption

1. Evaluate char. poly. of key
2. Reconcile with ciphertext
3. Reproduce string

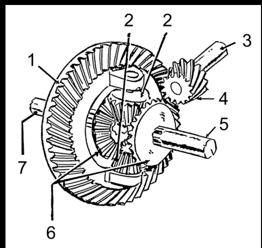
$\chi_{key}(Z)$= (Z-[mak])(Z-[e c]) (Z-[mix]) (Z-[cok])(Z-[e m]) …

S_p= {To_, o_m, mak, ake, ke_, e_c, _co, cok, oke, ke_, e_m, _mi, mix, …}

p= To make coke mix caffeine with citric acid and juice.

Ciphertext: $\chi_{S_p}(0)$, $\chi_{S_p}(15)$, $\chi_{S_p}(51)$, $\chi_{S_p}(60)$, $\chi_{S_p}(85)$, $\chi_{S_p}(90)$, $\chi_{S_p}(102)$, $\chi_{S_p}(105)$

Key: [mak, e_c, mix, cok, e_m]



Implementation

Deception

1. Evaluate char. poly. of *decoy* key
2. Reconcile with ciphertext
3. Reproduce *decoy* string

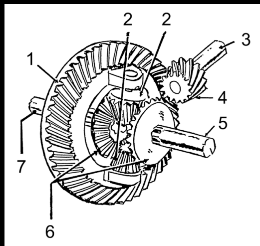
χ_{decoy}(Z)= (Z-[co])(Z-[coa])
(Z-[oal]) (Z-[al])(Z-[he]) …

S_{decoy}= {To , o m, mak, ake,
ke , e c,…, mi, mix, ix ,
x c, co, coa, oal, …}

p'= To make coke mix coal with heat and water.

Ciphertext: $\chi_{S_p}(0), \chi_{S_p}(15), \chi_{S_p}(51), \chi_{S_p}(60), \chi_{S_p}(85), \chi_{S_p}(90), \chi_{S_p}(102), \chi_{S_p}(105)$

Decoy key: `[_co, coa, oal, al_, _he, hea, eat, at_]`



Implementation – *Example*

Snippets from *Hackers, Heroes of the Computer Revolution by* Stephen Levy

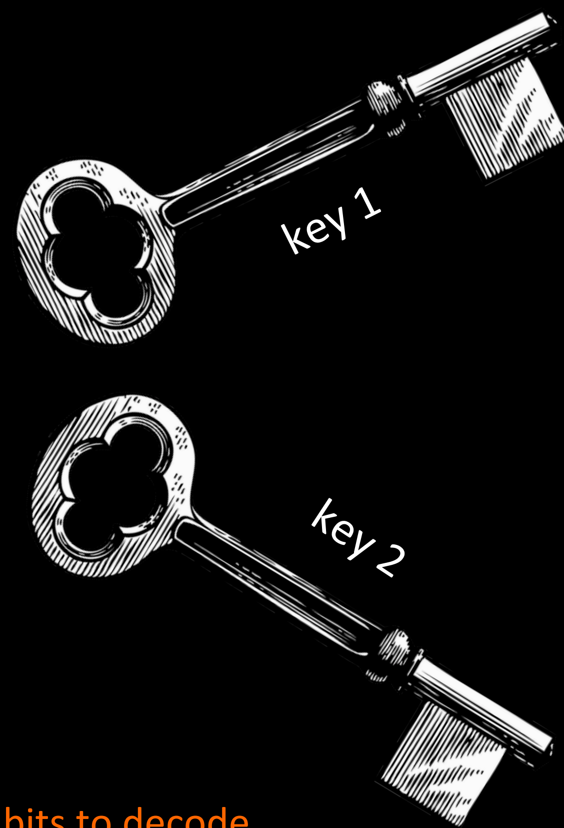
Text Size	Cipher length	Key length
1280	70687	1001
2560	144907	1287
5120	293587	1001
10240	591307	1241
20480	1194188	1089
40960	2401208	1361

Two texts differing by one burst of 5 edits

Conclusion

… where we got tired of thinking

- One encryption, many plaintexts
  - plan-ahead
  - after encryption
  - incremental
-



- Implementation
  - set reconciliation
  - string reconciliation
  - unique decoding
- Security
  - information-theoretic – need ~log(n) secret bits to decode
  - exist valid decoy texts - may not be meaningful

Conclusion

Limitations:

(n-character plaintext)

- *Slow encryption*: ~$n^2 \log n$
- *Slow decryption*: ~n^3
- *High overhead*

Considerations:

- Information in keys
- Key shortening

Extensions

- Other coding
- Encryption in blocks

References

Applications

- Truecrypt: Free open-source on-the-fly encryption. truecrypt.org
- Andrew D. McDonald and MarkusG. Kuhn. Stegfs: A steganographic file system for linux. In Andreas Pfitzmann, editor, *Information Hiding*, volume 1768 of *Lecture Notes in Computer Science*, pages 463–477. Springer Berlin Heidelberg, 2000.

Set reconciliation

- Y. Minsky, A. Trachtenberg, and R. Zippel. Set reconciliation with nearly optimal communication complexity. *IEEE Trans. on Info. Theory*, September 2003.

String reconciliation

- Sachin Agarwal, Vikas Chauhan, and Ari Trachtenberg. Bandwidth efficient string reconciliation using puzzles. *Parallel and Distributed Systems, IEEE Transactions on*, 17(11):1217–1225, 2006.
- Arnold Filtser, Jiaxi Jin, Aryeh Kontorovich, and Ari Trachtenberg. Efficient determination of the unique decodability of a string. In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pages 1411–1415. IEEE, 2013.
- J. Jin. Prioritized data synchronization with applications. Master's thesis, Boston University, 2012.
- Aryeh Kontorovich and Ari Trachtenberg. Deciding unique decodability of bigram counts via finite automata. *Journal of Computer and System Sciences*, 2013.

Crypto

- Ross Anderson, Roger Needham, and Adi Shamir. The steganographic file system. In *Information Hiding*, pages 73–82. Springer, 1998.
- Ran Canetti, Cynthia Dwork, Moni Naor, and Rafail Ostrovsky. Deniable encryption. In *Advances in Cryptology-CRYPTO'97*, pages 90–104. Springer, 1997.
- Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.
- Marek Klonowski, Przemysaw Kubiak, and Mirosaw Kutyowski. Practical deniable encryption. In Viliam Geffert, Juhani Karhumki,
- Alberto Bertoni, Bart Preneel, Pavol Nvrat, and Mria Bielikov, editors, *SOFSEM 2008: Theory and Practice of Computer Science*, volume 4910 of *Lecture Notes in Computer Science*, pages 599–609. Springer Berlin Heidelberg, 2008.
- Adam O'Neill, Chris Peikert, and Brent Waters. Bi-deniable public-key encryption. Cryptology ePrint Archive, Report 2011/352, 2011. http://eprint.iacr.org/.