MUNI
FI

Biased RSA private keys

Origin attribution of GCD-factorable keys

Adam Janovsky, Matus Nemec, Petr Svenda, Peter Sekan, Vashek Matyas
adamjanovsky@mail.muni.cz

Center for Research on Cryptography and Security, Faculty of Informatics, Masaryk University, Czech Republic

October 5, 2020

Imagine RSA public key

n = 9782D7123C330444C88E279BF321EE84AC39524F1D8402632
7B04F32E1E930FC81588010178DC75FCBF8258A068071317245D0
8817988813C4173495A922A41DA429A964F738020076EFFE7ED58
11088873C6E58EEF1CDC900596681F490BE72368B51A821FC699E
9C3FD66B377E2DF2485DC401DD99CC125890E5D969A6AC8B
e = 10001

Imagine RSA public key

n = 9782D7123C330444C88E279BF321EE84AC39524F1D8402632
7B04F32E1E930FC81588010178DC75FCBF8258A068071317245D0
8817988813C4173495A922A41DA429A964F738020076EFFE7ED58
11088873C6E58EEF1CDC900596681F490BE72368B51A821FC699E
9C3FD66B377E2DF2485DC401DD99CC125890E5D969A6AC8B
e = 10001

What source generated this key?

OpenSSL

Imagine RSA public key

n = 9782D7123C330444C88E279BF321EE84AC39524F1D8402632
7B04F32E1E930FC81588010178DC75FCBF8258A06807131 7245D0
8817988813C4173495A922A41DA429A964F738020076EFFE7ED58
11088873C6E58EEF1CDC900596681F490BE72368B51A821FC699E
9C3FD66B377E2DF2485DC401DD99CC125890E5D969A6AC8B
e = 10001

What source generated this key?

OpenSSL

Infineon

RSA Primer

$$n = p \cdot q$$

Back into 2016 and 2017

- Svenda et. al identified several sources of bias in public keys [4].

Back into 2016 and 2017

- Svenda et. al identified several sources of bias in public keys [4].
- They built a model capable of recognizing 13 groups of keys with accuracy 40%.

Back into 2016 and 2017

- Svenda et. al identified several sources of bias in public keys [4].
- They built a model capable of recognizing 13 groups of keys with accuracy 40%.
- In 2017, their work resulted into ROCA vulnerability: Practically factorable keys found in millions of devices [3].

Back into 2016 and 2017

- Svenda et. al identified several sources of bias in public keys [4].
- They built a model capable of recognizing 13 groups of keys with accuracy 40%.
- In 2017, their work resulted into ROCA vulnerability: Practically factorable keys found in millions of devices [3].
- But what about private keys?

Scenarios with private keys

- Personal scrutiny.

Scenarios with private keys

- Personal scrutiny.
- Company audits.

Scenarios with private keys

- Personal scrutiny.
- Company audits.
- Forensic investigation of factored keys from unknown source.

Sources of Bias in RSA keys

1. Performance optimizations.

Sources of Bias in RSA keys

1. Performance optimizations.
2. Type of primes: probable, strong, and provable primes.

Sources of Bias in RSA keys

1. Performance optimizations.
2. Type of primes: probable, strong, and provable primes.
3. Ordering of primes: are the RSA primes in private key ordered by size?

Sources of Bias in RSA keys

1. Performance optimizations.
2. Type of primes: probable, strong, and provable primes.
3. Ordering of primes: are the RSA primes in private key ordered by size?
4. Proprietary algorithms.

Illustration of Bias

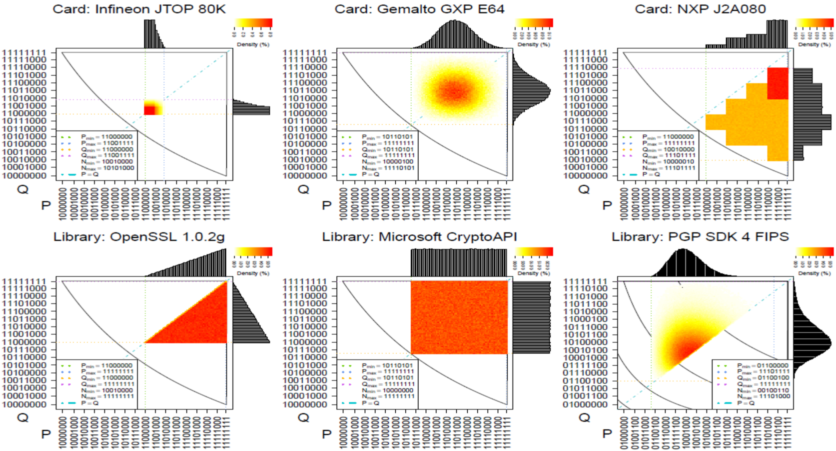

Figure: Distribution of MSBs in p, q of various libraries.

Attribution process

1. Collect many RSA keys.
2. Extract features $\rightarrow$ discover classes.
3. Build a model.
4. Evaluate the model on a test set.
5. Use GCD to factorize keys from the IPv4 wide scans.
6. Attribute the factorized keys.

Bias representatives

1. 5MSB of p, q
2. Blum primes
3. Small divisors of $p - 1$ and $q - 1$ avoided up to the value: 17683, or 251, or 5, or not at all.
4. ROCA fingerprint.

Class discovery

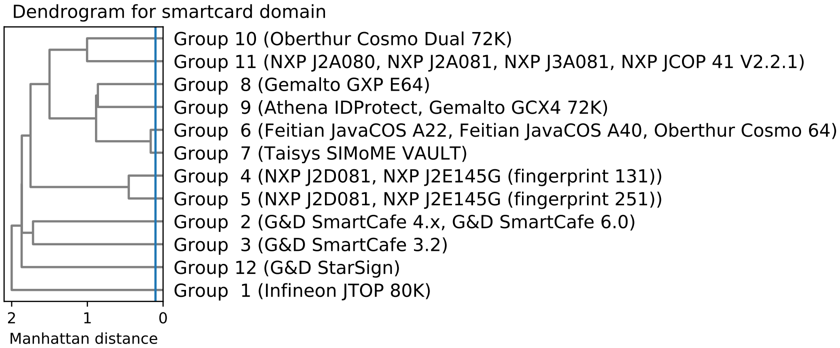

Figure: Dendrogram of sources from smartcards domain.

Model selection and evaluation

- The feature set is quite narrow

Model selection and evaluation

- The feature set is quite narrow $\rightarrow$ enables for Bayes Classifier.

Model selection and evaluation

- The feature set is quite narrow $\rightarrow$ enables for Bayes Classifier.
- Bayes classifier: Table of empirical probabilities.
  - Training: For each possible feature vector count the most probable source.
  - Test time: When $((251, 251), \mathrm{False}, 5, \mathrm{False})$ encountered, find the most probable source.

Model selection and evaluation

- The feature set is quite narrow $\rightarrow$ enables for Bayes Classifier.
- Bayes classifier: Table of empirical probabilities.
  - Training: For each possible feature vector count the most probable source.
  - Test time: When $((251, 251), \text{False}, 5, \text{False})$ encountered, find the most probable source.
- Dataset: 157 million of training keys, 1.8 million of test keys.

Model selection and evaluation

- The feature set is quite narrow $\rightarrow$ enables for Bayes Classifier.
- Bayes classifier: Table of empirical probabilities.
  - Training: For each possible feature vector count the most probable source.
  - Test time: When $((251, 251), \mathrm{False}, 5, \mathrm{False})$ encountered, find the most probable source.
- Dataset: 157 million of training keys, 1.8 million of test keys.
- All domains: 26 groups, 47% accuracy, 3 groups with 100% accuracy.

GCD-factorable keys

- As of 2016, almost 1% of RSA keys in TLS certificates were practically factorable.
- Discovered in 2012 independently by Lenstra et al. [2] and Henninger et al. [1].

GCD-factorable keys

- As of 2016, almost 1% of RSA keys in TLS certificates were practically factorable.
- Discovered in 2012 independently by Lenstra et al. [2] and Henninger et al. [1].
- Reason: Insufficient entropy after boot $\rightarrow$ shared prime.

GCD-factorable keys

- As of 2016, almost 1% of RSA keys in TLS certificates were practically factorable.
- Discovered in 2012 independently by Lenstra et al. [2] and Henninger et al. [1].
- Reason: Insufficient entropy after boot $\rightarrow$ shared prime.
- We used Rapid7 dataset to obtain scans from October 2013 - July 2019.

GCD-factorable keys

- As of 2016, almost 1% of RSA keys in TLS certificates were practically factorable.
- Discovered in 2012 independently by Lenstra et al. [2] and Henninger et al. [1].
- Reason: Insufficient entropy after boot $\rightarrow$ shared prime.
- We used Rapid7 dataset to obtain scans from October 2013 - July 2019.
- Assumption: When a batch of GCD-factored keys shares a prime, they were all generated by sources from a single classification group.

GCD-factorable keys

- As of 2016, almost 1% of RSA keys in TLS certificates were practically factorable.
- Discovered in 2012 independently by Lenstra et al. [2] and Henninger et al. [1].
- Reason: Insufficient entropy after boot $\rightarrow$ shared prime.
- We used Rapid7 dataset to obtain scans from October 2013 - July 2019.
- Assumption: When a batch of GCD-factored keys shares a prime, they were all generated by sources from a single classification group.
- Apart from the OpenSSL, origin of the factorable keys is unknown.

How reliable are our results?

Number of primes in a batch	1	10	20	30	100
Group 1	100.0%	100.0%	100.0%	100.0%	100.0%
Group 2	42.8%	99.7%	100.0%	100.0%	100.0%
Group 3	78.0%	100.0%	100.0%	100.0%	100.0%
Group 4	47.5%	90.3%	95.8%	98.7%	100.0%
Group 5|13	1.8%	30.8%	43.7%	51.8%	74.7%
Group 6	5.2%	48.9%	61.0%	64.8%	76.7%
Group 7|11	0.0%	67.3%	92.3%	97.4%	100.0%
Group 8|9|10	37.9%	99.9%	100.0%	100.0%	100.0%
Group 12	12.8%	61.8%	77.7%	83.9%	97.2%
Average	**36.2%**	77.6%	85.6%	88.5%	94.3%

Figure: Accuracy of model on GCD-factorable keys.

Sources of GCD-factorable keys

- 82 thousand primes in 2511 batches.
- 2230 batches (88%) from OpenSSL (well matches previous research).
- 3 batches from 8-bit OpenSSL.
- 278 batches (11%) from: Libgcrypt, Libgcrypt FIPS, OpenSSL FIPS, WolfSSL, SafeNet, cryptlib, Botan, LibTomCrypt, Nettle 3.2, Nettle 3.3.
- None from other 6 groups that cover 13 sources. These are very improbable sources of keys.

Conclusions

- Looking at private keys makes classification much more precise. From 40% accuracy on 13 groups (public keys) to 47% of accuracy on 26 groups.

Conclusions

- Looking at private keys makes classification much more precise. From 40% accuracy on 13 groups (public keys) to 47% of accuracy on 26 groups.
- Our models are especially reliable when on limited domain or batch of keys is available.
- For instance, 10 keys $\rightarrow$ 89% accuracy (4% random guess) on 26 groups.

Conclusions

- Looking at private keys makes classification much more precise. From 40% accuracy on 13 groups (public keys) to 47% of accuracy on 26 groups.
- Our models are especially reliable when on limited domain or batch of keys is available.
- For instance, 10 keys $\rightarrow$ 89% accuracy (4% random guess) on 26 groups.
- Real-world use-cases of private key classification exist.

Visit our website

`https://crocs.fi.muni.cz/public/papers/privrsa_esorics20`

References

[1] Nadia Heninger et al. “Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices”. In: *Proceeding of USENIX Security Symposium*. USENIX, 2012, pp. 205–220.

[2] Arjen K. Lenstra et al. *Ron was wrong, Whit is right*. Cryptology ePrint Archive, Report 2012/064. [cit. 2020-07-13]. Available from `https://eprint.iacr.org/2012/064`. 2012.

[3] Matus Nemec et al. “The Return of Coppersmith’s Attack: Practical Factorization of Widely Used RSA Moduli”. In: *24th ACM Conference on Computer and Communications Security (CCS’2017)*. ACM, 2017, pp. 1631–1648.

[4] Petr Svenda et al. “The Million-Key Question – Investigating the Origins of RSA Public Keys”. In: *Proceeding of USENIX Security Symposium*. 2016, pp. 893–910.

MUNI
FACULTY
OF INFORMATICS