Usable everlasting encryption using the pornography infrastructure (Fast Abstract)

Enka Blanchard
Digitrust, Loria, Université de Lorraine
Nancy, France
enka.blanchard@gmail.com

Siargey Kachanovich
Corpy&Co., Inc.
Tokyo, Japan

***Abstract*—Nine years before Snapchat and its ephemeral messages, Aumann, Ding, and Rabin introduced the idea of everlasting security: an encryption that could not be decrypted after a certain date, no matter the adversary's computing power. Their method is efficient but not adapted to real-life constraints and cannot effectively be used today.**

In this paper we look at potential entropy sources available today, and propose a new solution that makes use of the already existing communications from pornography distribution networks, and look at its social implications. The method proposed has multiple advantages stemming from the fact that pornography is shameful in most societies, and it is usable off-the-shelf by individuals with limited technical skills, although it still requires some effort.

***Index Terms*—Usable security, Random beacon, Bounded memory model, One-time pad**

Previous work

Aumann, Ding and Rabin introduced the first everlasting encryption algorithm in 2002 [1], where, a bounded time after the encryption, even an adversary with unbounded computational capabilities only has an exponentially small probability of being able to decrypt the message. This assumes two elements: a common source of public random bits, and a bound on the adversary's capacity to store the random bits. Finding a common source of public random bits with very high throughput, however, reveals to be challenging. The original paper mentions compressing large amounts of online text, or using a constellation of satellites that beam random bits at a high rate. The second is too costly to be usable by the average user, and the first creates multiple problems, detailed below.

Model and algorithm

Let's suppose Alice wants to send secret data to Bob, without Alice's supervisor Eve finding out. Eve can intercept all messages from her agents (among which is Alice), and store all the encrypted ones for potential later decryption, but can only detect patterns and store a fraction of the information exchanged in clear because of size constraints.

We can use an algorithm close to the original protocol [1]. Using asymmetric encryption — which we can initially assume to be secure for at least a limited time — Alice sends a set of pseudo-random numbers to Bob. Those are pointers to a common source of public random bits. After they confirm having the same data — e.g. by the use of a checksum — Alice and Bob can exchange using the equivalent of a one-time pad. As long as Eve does not store the whole data required, she will not be able to decrypt Alice's messages. The hard task is then to find a good and discreet source of random bits which does not trigger Eve's patterns. Alternatively, a public source of low-entropy bits can also work as long as a randomness extractor is used, for example the same extractor used in the original paper [1].

Entropy sources

There are three properties that we require from our entropy sources. First, *throughput* should be high, with 1 TB/s is a good initial target to counter most adversaries. Second, there should be a way to construct *canonical* pointers to the data stream. Third, Alice's data access should be statistically indistinguishable from her colleagues'.

Of the two sources investigated in the original paper, the satellite-based random beacons are not financially feasible today. The random web-page compression fails for simple throughput reasons (if we discard multimedia content). One could instead use the NIST randomness beacon [2]), which was not yet in service, but this source only produces 512 bits per minute today. Even with increased throughput, Eve could store all the random bits requested from the beacon. A more realistic solution would be to use online video traffic, representing more than half of the 500TB/s of internet throughput [3], especially hard to compress non-animated videos [4].

Downstream traffic is generally too redundant. Youtube, for example, sends more than 50TB/s from its servers to many devices, but its total size only increases at a rate of 40GB/s [5]. *Upstream* traffic then seems best with multiple many-to-many video streaming services, of which we consider three main ones.

Twitch — a streaming platform with a focus on video-game streaming — has a high throughput, with more than 5% of all upstream traffic. However, the streams are potentially highly compressible, as it mostly consists of video-game live-stream from a few major video-games, whose Kolmogorov complexity can be extremely low.

Video chat also has a high throughput — at least 8% of upstream internet traffic — and is not easily compressible without high losses. However, this traffic is generally not ac-

cessible to people outside the call, and it is highly distributed, making a canonical index difficult.

The last candidate we analyse is live pornography. Its real throughput is hard to estimate, but is often approximated at a few percent of the total throughput, and the largest live pornography web site (livejasmin.com) is consistently ranked in the top 50 most visited web sites worldwide. This site, which is just one of many sources, already has enough official streams to be in the tens of GB/s, similar to Youtube. However, its real throughput is at least one order of magnitude higher, as those streams create public answer streams. It is relatively accessible, despite potential financial and legal caveats (which we will cover below). The main issue is to define a canonical index, which we will now tackle.

Protocol

A detailed protocol is available in the full version of this paper. It requires Alice to send n pointers to Bob and for both to record at least $n-1$ streams. Each pointer contains five elements: the URL of a web site, the duration of the video to record, the index of a stream on that web site, the time to start recording the stream, and a nonce to coordinate the two recordings. The difficulty is in agreeing on the stream index and frame index.

We propose the following solution for the stream: Alice sends a large number x and a small number c. If the web site has k streams when Alice accesses it, she selects the stream number $\left(x \bmod \lfloor \frac{k}{c} \rfloor\right)$. This makes the probability of Alice agreeing with Bob on the stream in $\Theta(\frac{1}{c})$. If we want to look at response streams, this method can be used recursively.

For the starting frame, Alice sends a nonce of the appropriate precision, and hashes frames until one of them agrees with the nonce, and uses that as a starting frame.

With a canonical stream, a set duration and a starting frame, Alice and Bob can both extract entropy from the stream. By setting $n = 10$ and sending a single parity stream, we already have good guarantees. Eve needs to be lucky and store at least 9 out of the 10 streams, as she can only store a constant fraction of the world's video streams. Even if this fraction is 10% of all streams, her probability of having enough information to decrypt the message is still at most

$$10 \times \frac{9}{10} \times \left(\frac{1}{10}\right)^9 + \left(\frac{1}{10}\right)^{10} \approx 9 \times 10^{-9}.$$

Social aspects and discussion

One important limitation of this protocol is that it relies on Alice and Bob having access to live pornography. This requires it to be legal (which makes using it impossible to use in China and difficult in Indonesia). It is true that Alice might invite additional scrutiny by looking at pornography from work, but two factors limit this. First, this behaviour is far more widespread than most people believe [6]. Second, the amount of pornography needed is in fact pretty limited. Assuming wide margins, encrypting 100MB of data would require about 150MB of pornographic videos, which could be obtained in a few minutes.

Using live pornography also has a few advantages. First, the fact that accessing pornography is frowned upon gives Alice plausible deniability for her suspicious behaviour. Getting caught accessing it would probably lead to a disciplinary hearing, but not for data exfiltration charges. Moreover, it is extremely frequent, with the majority of people admitting accessing pornography from their office . Second, large public investments to store live pornography would be hard to defend politically, even if it is for cybersecurity purposes.

The protocol can be used directly, without advanced tools or the creation of a large source of entropy. However, it requires a proof-of-concept, and multiple extensions are imaginable, such as: handling encoding variability in the source, improving stream agreement, increasing the fault tolerance, or addressing secretly compromised encryption methods.

For countries where the access to pornography is limited, there are alternatives that satisfy the constraints presented in this paper and could be used instead of our protocol. For example, we could use slightly altered data from P2P networks, or modify the Scuttlebutt protocol [7]. Contrary to the system we propose, these alternatives are not directly employable today, as they require the cooperation of a large set of users. Moreover, they would require a higher technical ability to implement. The increased traffic from IoT might also create new sources of public streams, and recent trends in the relative hardware costs for data storage compared to global throughput are currently working in our advantage, making other entropy sources more secure. Twitch is the main option, but the average compressibility of its feeds should be evaluated before it is used.

Acknowledgements

A complete version of this paper with agent models and explicit algorithms is available online on the first author's website and on HAL: https://hal.archives-ouvertes.fr/hal-02556366v2.

This work was supported by the French PIA project "Lorraine Université d'Excellence", ANR-15-IDEX04-LUE.

REFERENCES

[1] Y. Aumann, Y. Z. Ding, and M. O. Rabin, "Everlasting security in the bounded storage model," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1668–1680, 2002.

[2] M. J. Fischer, M. Iorga, and R. Peralta, "A public randomness service," in *Proceedings of the International Conference on Security and Cryptography – SECRYPT*. IEEE, 2011, pp. 434–438.

[3] Cisco, "Global visual networking index: Forecast and trends, 2017-2022," Cisco, Tech. Rep., 2018. [Online]. Available: https://web.archive.org/web/20190323025839/https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html

[4] K. Kawaharada, K. Ohzeki, and U. Speidel, "Information and entropy measurements on video sequences," in *2005 5th International Conference on Information Communications Signal Processing*, 2005, pp. 1150–1154.

[5] C. Cullen, "Global internet phenomena report," Sandvine, Tech. Rep., 2018.

[6] T. McDonald. (2018) How many people watch porn at work will shock you. [Online]. Available: https://web.archive.org/web/20180205170953/https://sugarcookie.com/2018/01/watch-porn-at-work/

[7] R. van Renesse, D. Dumitriu, V. Gough, and C. Thomas, "Efficient reconciliation and flow control for anti-entropy protocols," in *Proceedings of the 2nd Workshop on Large-Scale Distributed Systems and Middleware*, ser. LADIS '08. New York, NY, USA: ACM, 2008, pp. 1–7.