Hedged Public-Key Encryption: How to Protect Against Bad Randomness

ECS 227 Paper Writeup

Andy Applebaum

1 Introduction

The focus of this paper is to examine the security of encryption schemes in the face of "bad randomness": a good scheme with good randomness should achieve a strong sense of security (IND-CPA), or, given bad randomness, should still achieve a somewhat strong sense of security (PRIV). The authors identify this by defining a new notion of security, "Indistinguishability under a Chosen Distribution Attack" (IND-CDA). Put plainly, unlike an IND-CPA attack, where the adversary gets to choose the plaintexts, in an IND-CDA attack, the adversary gets to specify the distribution of the plaintexts as well as the "random" behaivor of the encryption function; the goal of the adversary is, like with IND-CPA, to distinguish left-or-right encryption. With complete freedom over the distributions, it would be trivial for the adversary to achieve this, so the authors place small requirements on the adversary's distributions, most of the ideas stemming from "min-entropy". The authors provide a few constructions that can achieve the IND-CDA notion of security, considering primarily non-adaptive attacks. Towards the end of the paper, the authors introduce an "anonymity" concept that takes into account the idea of distribution-choosing, ultimately showing that this concept combined with non-adaptive IND-CDA results in adaptive IND-CDA.

2 Background and Definitions

One of the first notions they present is the idea of "hedging". Essentially, they want a "good" scheme to still be good when given good randomness, and, when given bad randomness instead, maybe not be good, but at least be okay. They encompass this idea with the security notion of H-IND (hedged security), which refers to schemes that are secure both in the IND-CPA sense and in the IND-CDA sense.

Let $\mathcal{S} = \{\mathcal{P}, \mathcal{K}, \mathcal{E}, \mathcal{D}\}$ be a public-key encryption scheme (PKES) with length parameter $n(\cdot)$ and:

- $\mathcal{P}$ the parameter generation algorithm, which, given 1^k, outputs a parameter string par.
- $\mathcal{K}$ the key generation algorithm, which, given a parameter string par, outputs a key pair (pk, sk) (pk the public key and sk the secret key).
- $\mathcal{E}$ the encryption algorithm, which, given the public key pk and message $m \in \{0,1\}^{n(k)}$, returns the encryption of m: $\mathcal{E}(pk, m)$.
- $\mathcal{D}$ the decryption algorithm, which, given the secret key sk and ciphertext c, returns the appropriate message or $\perp$.

For $\mathcal{S}$ to be a secure encryption scheme, we must have that $\mathcal{E}$ is a randomized algorithm (so, $\mathcal{E}(pk, m)$ is not always the same when called twice). Instead of letting $\mathcal{E}$ be randomized, we can instead re-write it as a deterministic algorithm that takes an additional "advice" string that corresponds to random coin tosses. This leads to the following new definition for a PKES: $\mathcal{S}_d = \{\mathcal{P}, \mathcal{K}, \mathcal{E}_d, \mathcal{D}\}$ with length parameter $n(\cdot)$, randomness length parameter $\rho(\cdot)$, and where $\mathcal{E}_d$ is run the same as $\mathcal{E}$, but also takes in an $r \in \{0,1\}^{\rho(k)}$ that corresponds to choices (i.e., coin flips) for its transition function. Unlike $\mathcal{S}$, $\mathcal{S}_d$ is, to some degree, deterministic.

Because a "distribution attack" allows the adversary to specify the distribution, the r above loses its generality, and $\mathcal{S}_d$ becomes more and more deterministic. In order to obtain strong definitions for IND-CDA, they reference another security notion, PRIV, which is defined for deterministic encryption schemes. The authors do not define PRIV, insted citing "Deterministic and Efficiently Searchable Encryption" (Bellare, Boldyreva, O'Neill) as a reference. Looking at that paper, in a very broad sense, a scheme is PRIV-secure if the ciphertext of a message leaks very little about the plaintext of that message. The PRIV game/experiment is, in a very loose sense, run in three steps: one adversary creates a message and some string that contains information about it, the oracle encrypts that message, and then another adversary tries to guess what the information about the message is, just given the ciphertext of it.

Underlying the ideas of randomness, distributions, and even in the formal definition of PRIV, is the concept of “min-entropy”. This concept does not originate from this paper – they reference “Entropic Security and the Encryption of High Entropy Messages” (Dodis, Smith), where min-entropy is defined over a random variable A as “a measure of uncertainty of its outcome”, with the more quantitative formula given:

$$H_\infty(A) = -log(\max_a \Pr(A = a))$$

This paper does not use the above formula directly, instead providing their own definition, summarized as follows: a t-source is a probabilistic algorithm $\mathcal{M}$ with message length $n(\cdot)$ and randomness length $\rho(\cdot)$ that on input 1^k returns t messages $(m_1, ..., m_t)$ and a string r where each $m_i \in \{0,1\}^{n(k)}$ and $r \in \{0,1\}^{\rho(k)}$. Then, $\mathcal{M}$ has min-entropy $\mu(\cdot)$ if:

$$\Pr[(m_b, r) = (x, y)] \leq 2^{-\mu(k)}$$

for all $k \in \mathbb{N}$, $b \in \{1, ..., t\}$, $x \in \{0,1\}^{n(k)}$ and $y \in \{0,1\}^{\rho(k)}$. Combining this definition with the one from the other paper yields a simplified text-version: essentially, the “random variable” A is the output of $\mathcal{M}$, which is checked against equality for each (x, y) pair – the most likely of these to be true corresponds to the min-entropy. So, a “random” scheme that returns either $(0, 0)$ or $(1, 0)$ would have high min-entropy as the likelyhood that its output was, for example, $(1, 0)$ is high. Even as k grows large and the space for (x, y) grows, the min-entropy essentially remains the same as it takes the maximum likelihood over all possible (x, y) pairs.

3 Defining IND-CDA

They provide a precise definition of $CDA_{\mathcal{S},k}$ by using the game reproduced below:

```
proc. Initialize($1^k$):
    $par \xleftarrow{\$} \mathcal{P}(1^k)$
    $(pk, sk) \xleftarrow{\$} \mathcal{K}(par)$
    $b \xleftarrow{\$} \{0,1\}$
    Ret $par$

proc. LR($\mathcal{M}$):
    if pkout = true
    then
        Ret $\perp$
    else
        $(m_0, m_1, r) \xleftarrow{\$} \mathcal{M}(1^k)$
        Ret $\mathcal{E}(pk, m_b, r)$

proc. RevealPK():
    pkout $\leftarrow$ true
    Ret $pk$

proc. Finalize($b'$):
    Ret $(b = b')$
```

They require that the adversary, A, make queries (to **LR**) as an “mmr-source”, which is a 2-source as defined alongside min-entropy with $\rho > 0$, and give the CDA advantage as:

$$\mathrm{Adv}^{cda}_{\mathcal{S},A}(k) = 2 \cdot \Pr[\mathrm{CDA}^{A}_{\mathcal{S},k} \Rightarrow \mathrm{true}] - 1$$

This definition defines more than just the randomness of $\mathcal{E}$ – it also provides randomness to the message space. They justify this by explaining that in a good scheme there really should be some min-entropy in the message space, otherwie it would be too easy to brute-force the method. Additionally, by providing some randomness in the message space, they “weaken” the requirement on the randomness: they claim that security for IND-CDA is guaranteed so long as the *joint* (as in, the combination of) distribution of the message and randomness has high min-entropy. This highlights one of the differences between the game for IND-CPA, where the adversary chooses messages directly, and IND-CDA, where the adversary specifies a distribution over the message space. To put it bluntly, A's queries to **LR** are probabilistic algorithms; in essence, A is giving an oracle to its oracle.

One other key distinction between IND-CDA and IND-CPA is the revelation of the public key. Since $\mathcal{S}_d$ is in some regards deterministic (or, at least, A can control it's randomness), if A knows *pk*, then A can trivially encrypt messages and always win the game. Thus, in the game above, A cannot learn the public key and continue to make LR queries. On a similar note, because of the pseudo-determinism of $\mathcal{S}_d$, there may be equality-attacks (by querying the same or similar $\mathcal{M}$'s twice), and they provide a restriction to guarantee against this. Finally, they acknowledge that unlike IND-CPA, IND-CDA does not guarantee adaptive security given non-adaptive security; for the majority of their paper, they assume non-adaptive security notions, only at the end providing a proof for an adaptive system.

4 Constructions

They provide a few constructions and select proofs of their security. Perhaps the most straightforward scheme is "pad-then-deterministic" (PtD). For PtD, given a deterministic encryption scheme, they want to construct a randomized encryption scheme: they do this by holding $\mathcal{P}$ and $\mathcal{K}$ constant, and modifying $\mathcal{E}_r(pk_d, m, r) = \mathcal{E}_d(pk_d, r||m)$, changing $\mathcal{D}$ to remove the concatenated r as needed. This scheme is non-adaptive H-IND secure – provided that the underlying deterministic scheme is PRIV-secure, then, because the $\mathcal{M}$ queries need to have sufficient min-entropy, the resulting $\mathcal{E}_r$ should have sufficient min-entropy, showing IND-CDA security. From there, proving IND-CPA is more difficult, and they cite another version of this paper to do so.

Another construction is "randomized-then-deterministic" (RtD). This scheme works as the name suggests, first using S_r to encrypt the message, and then using S_d to encrypt the resulting ciphertext concatenated with some padding. The resulting scheme is non-adaptive H-IND secure; in the interest of brevity, the proof is omitted. Interestingly, the reverse method, "deterministic-then-randomized" (DtR), is not H-IND secure. They do not provide a proof, however, the idea is that there will not be a guarantee that the join distribution of messages and randomness has sufficient min-entropy; it seems that even if $\mathcal{S}_r$ is IND-CPA secure, it doesn't guarantee good min-entropy over the randomness, so even if $\mathcal{S}_d$ provides good min-entropy over the messages, the end result is an output of S_r, so $\mathcal{S}_r$'s low entropy is more influential.

5 Anonymity and Adaptive IND-CDA

The authors present a security notion that combines anonymity with the IND-CDA ideas. The basic idea is that a secure scheme with regards to anonymity ensures that ciphertexts don't leak information about the public key that's used. The actual game presented for ANON is not reproduced, however the idea is fairly straightforward: the game initializes with two pairs of public/secret keys, and then allows the adversary to encrypt queries using one of them. These queries are similar to the game for IND-CDA in that the adversary is specifying a t-source, this time a 1-source. After running all the encryption queries, the adversary is given a challenge that's the ciphertext for some pair returned by $\mathcal{M}$ and both keys; the adversary wins if it correctly identifies the key used. They use this notion of ANON security to show that if some scheme is non-adaptive IND-CDA secure as well as ANON secure, then it is adaptive CDA secure (the proof is omitted in this version of the paper).

6 Discission/Conclusions

The chosen-distribution-attack has strong prospects with regards to applicability (they recognize failures of randomization in the introduction), and creates an interesting middleground between randomized encryption schemes and deterministic ones. The results of the paper seem strong and to some extent can be viewed as a generalization of other ideas (i.e., a randomized secure scheme will have high min-entropy, while a deterministic one has low min-entropy). While the topic of bad randomness in schemes is already present in the literature, the authors distinguish their paper by incorporating the message-entropy into the overall randomness of the algorithm, giving it a slightly different construct.

Unfortunately, there seem to be a few loose ends in the paper – with regards to that last idea, the relation to other work feels slightly incomplete. Namely, do other papers that deal with bad randomness say anything with regards to message randomness? How do the IND-CDA notions of this paper compare to the other notions in the related papers? On a similar note, the inherited definitions are not explicitly stated; PRIV, for example, is never fully defined. While a formalization of min-entropy is presented, it feels vague without the simple line from the referenced paper. Throughout the paper, "high" min-entropy and "low" min-entropy are never truly defined – combining this problem with the somewhat unclear min-entropy definition makes some of the new concepts harder to understand.

One question that feels unanswered is in the design of $\mathcal{M}$: in the definition of **LR**, when **LR** queries $\mathcal{M}$, does A know the response of $\mathcal{M}$? If A knows any of the tuples returned by $\mathcal{M}$, then IND-CDA is easy to break: if, on query q, A knows $(m_0, m_1, r) \overset{\$}{\leftarrow} \mathcal{M}(1^k)$, then on the next query, A simply calls **RevealPK()** and then manually encrypts m_0 and m_1, comparing both to the c returned by **LR**. While the ability to choose the distribution does not imply the ability to choose the actual values, it isn't clear that a min-entropy $\mathcal{M}$ guarantees that A will never know the output of $\mathcal{M}$; thinking out loud, it seems to imply this, but without precise definitions for high/low min-entropy, it's not easy to conclude this. As an extension of this confusion, their proof of how equality can cause problems requires $\mathcal{M}$ to output something to the effect of (a, a, r) and then (a, a', r) – this idea seems to imply that not only does A know the output of $\mathcal{M}$, but A is directly controlling it.

One possible extension of this idea would be to see if it would be possible to give the adversary control over the distribution of keys and whether that effected indistinguishability or anonymity security. This idea may in fact be trivial, but, as an extension of the ideas in this paper, it would be interesting to see schemes that aren't defined as "random" or "deterministic" but rather "random-with-entropy", and having more generic security terms that correspond to that entropy.