Randomness in Cryptography Febuary 25, 2013

Lecture 8: Cryptography in the presence of local/public randomness

Lecturer: Yevgeniy Dodis *Scribe: Hamidreza Jahanjou*

So far we have only considered weak randomness sources and derived many impossibility results especially with respect to privacy. Now we wish to examine a related setting in which only the secret key (SK) is weak but perfect local or public randomness is available. First, let's go over some motivating examples.

Motivation

The following examples serve as motivation for the study of the aforementioned settings.

1. Biometrics. In many applications, such as retina scan, fingerprint recognition, voice recognition or even long passwords, the secret key is unlikely to be drawn from a uniform distribution. An extra issue here is noise (more on this later).
2. Leakage. Side-channel attacks. One important issue here is adaptivity.
3. Public parameters published by trusted sources. We assume that these trusted sources have good randomness (CRS).
4. Perfect/local randomness produced by different party than the one holding or generating the secret key. One common situation is publc-key cryptography.
5. Quantum computation/cryptography.

Roughly speaking, there are two kinds of questions in this area one passive and one active:

Passive Questions such as

1. How does the security of a given application degrade with weak sources where perfect local/public randomness remains good?
2. Identify general classes of apps friendly to weak secret keys.

Active Questions such as

1. How to design applications resisting weak/leaky keys?
2. (General approach) how to design probabilistic randomness extractors using public randomness (seed)? Here, a crucial assumption is that X, the distribution of the key, is independent of the seed.

Our study has some side benefits as well. In particular, extractors have many other applications beyond key generation:

1. Derandomization of BPP and IP.
2. Tools for arguing security (a type of security known as entropic security).
3. Weak to strong FHE (fully homomorphic encryption).
4. Leakage-resilient encryption.
5. Deterministic extractors for some sources, ERFs (exposure-resilient functions).

It should be mentioned, however, that despite their generality and usefulness, the existence of extractors is still not enough for many settings where there are extra constraints. Some examples are

- **Biometrics** error correction.
- **Bounded Storage Retrieval** local computability.
- **Independence of source and seed** leakage-resilient cryptography, RNG, randomness condensers.
- **Authentication of seed** between two parties (the interactive case), to oneself in the future (the non-interactive case).
- **Reusability/Unlinkability** extraction of an unbounded amount of uncorrelated randomness.
- **Special extra properties** collision resistance, unverifiability.
- **Computational entropy** related to Goldreich-Levin theorem.
- **Low entropy settings** where entropy loss is too large.

Before focusing on the passive questions, let's answer the following question. Which applications benefit from local randomness even if the secret key is drawn from a uniform distribution?

As we will see, local randomness, esp. coupled with extractors, allows us to improve efficiency and security assumptions. In the rest of the lecture today, we'll see four examples.

Example 1: CPA-Secure Symmetric/Public-Key Encryption

Here we seek $\mathsf{Enc}_{\text{key}}(m_0) \simeq \mathsf{Enc}_{\text{key}}(m_1)$ even if the attacker $\mathcal{A}$ can generate/ask for encryption of other messages including m_0 and m_1. This is not possible if the encryption depends only on SK/PK and is other wise deterministic; hence, a CPA[1]-encryption must be probabilistic.

Question 1 *How secure is CPA SKE/SKE with weak SK but perfect local randomness?*

[1] Chosen Plaintext Attack

Example 2: CCA (Chosen Cipher Attack) Encryption

We already know that the encryptor needs local randomness. However, here we claim there are advantages for the decryptor to have local randomness as well.

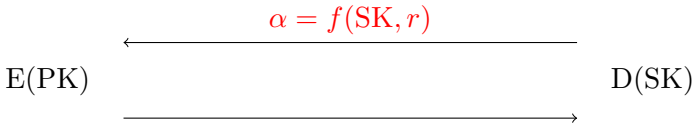
The idea behind forward security is to reduce the risk of exposure of SK. It is based on the key evolution paradigm. A user first registers a PK and keeps the corresponding secret key SK_0. The time during which the PK is expected to be valid is divided into periods and while the PK stays the same, there will be a sequence of SK's. In this way, if an SK is compromised, the past messages/transitions are still valid. Hence, forward security, is a means of keeping the authenticity of past messages in the even of a secret key exposure.

Getting CCA from CPA is the main feasibility problem in public-key encryption. There is also a weaker notion of q-CCA, where ¡= q decryption queries can be done. This can be done starting with a CPA, but only for a bounded value of q.

Here an $\mathcal{A}$ has access to PK and oracle access to $\mathsf{Dec}_{\text{key}}$. Now, $\mathcal{A}$ claims that, given c^* he is able to distinguish between $\mathsf{Enc}(m_0)$ and $\mathsf{Enc}(m_1)$. The only restriction is that $\mathcal{A}$ can not ask $\mathsf{Dec}(c^*)$.

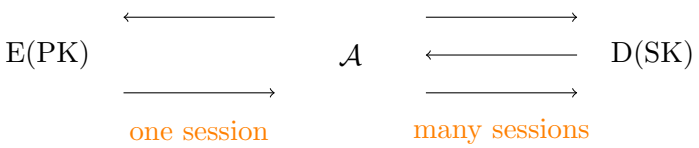
There are two possible issues with the above definition. First, there is no forward/replay security. Second, assumptions! We can do efficiently starting from most natural assumptions such as Factoring, DDH, etc. but not CPA.

Fortunately, there is a solution: interactive probabilistic decryption even with two rounds.



The second argument of f, r, denotes local randomness. Correctness is clear; let's focus on the security. To this end, we define here a generalization of CCA.

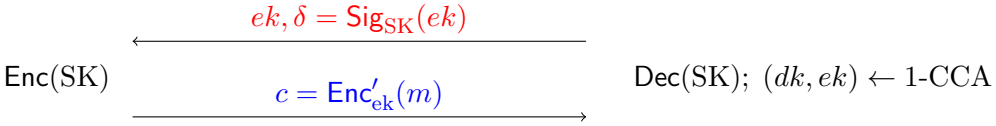
Definition 1 [iCCA] Consider an attacker $\mathcal{A}$ who has oracle access to one encryption session (i.e. at most one challenge) and any number of decryption sessions.



This is known as the interactive Chosen Cipher Attack. ♢

Theorem 1 *2-round ICCA exists if and only if CPA exists.*

Proof: The forward direction is left as an exercise. To prove the other direction, we use a 1-CCA scheme which follows from the CPA plus signature scheme which also follows from the CPA.



1-CCA is enough since D never reuses ek and it decrypts only once. Also note the this construction is both forward and replay secure.

□

Question 2 *How does this fare with weak SK but good local randomness?*

Example 3: Multi-time MACs $(\mathsf{Tag}, \mathsf{Ver})$

Here we have $\mathcal{A}^{\mathsf{Tag},\mathsf{Ver}} \to (m^*, \sigma^*)$ where $\mathcal{A}$ has oracle access to Tag and Ver.

DEFINITION 2 The advantage of $\mathcal{A}$ is defined as follows

$$\mathsf{Adv}_{\mathcal{A}} = \Pr[\mathsf{Ver}(m^*, \sigma^*) = 1 \text{ and } (m^*, \sigma^*) \text{ is fresh}].$$

♢

The traditional idea is $\sigma = f_s(m)$, where $\{f_s\}$ is a deterministic PRF function family. The issue here is that we need large-domain PRF or universal hash with large output or key.

Instead, the probabilistic idea is $\sigma = (r, f_s(r) \oplus h_t(m))$ where $sk = (s, t)$ and $\{h_t\}$ is an AUX[2] function family. The main advantage is that f has short input r and h hashes to output of f. It's cheaper to have h on a large domain and f on a small domain.

Note 1 *For probabilistic MAC, in general, access to* Ver *is essential.*

Lemma 1 *If* $\{f_s\}$*-PRF and* $\{h_t\}$*-AUX, then* $(\mathsf{Tag}, \mathsf{Ver})$ *construction above is secure.*

Proof Sketch: Consider the following games.

- **game 0** real $\mathcal{A}^{\mathsf{Tag},\mathsf{Ver}}$.
- **game 1** replace $f_s(\cdot)$ by truly random F. Note that this is OK by PRF security.
- **game 2** reject if collision occurs on r during $\mathsf{Tag}(\cdot)$. In this case $\Pr[\text{abort}] \leq O(q^2/2^{|r|})$.
- **game 3** replace $(r, F(r) \oplus h_t(m))$ by $(r, \$)$.
- **game 4** For every verification query (m_i, r_i, c_i) with new r reject.
- **game 5** For every verification query for old r reject. The point is that if $c_i = F(r) \oplus h_t(m_i)$ and $c_j = F(r) \oplus h_t(m_j)$ then $h_t(m_i) \oplus h_t(m_j) = c_i \oplus c_j$ where m_i and m_j are independent of t.

□

Another possible issue concerns assumptions/efficiency for MACs. Existing constructions from number-theoretic assumptions are inefficient. They are either "generic" tree-based (bit-by-bit) or essentially Naor-Reingold optimizations for DDH/Factoring. The problem is long keys.

[2] Almost XOR universal.

$$\mathsf{Tag}_{a_0,\ldots,a_n}(m_1,\ldots,m_n) = g^{a_0 \cdot \Pi_{i:m_i=1} a_i}$$

$|\text{key}| = O(\lambda \cdot |m|)$. We can play with hashing but there still will be a lower bound of $\Omega(\lambda^2)$.

Note 2 *It is possible to have DDH/Factoring/CDH?LPN constructions where* $|\sigma|, |s| = O(\lambda)$ *without a random oracle. One such construction is*

$$\begin{aligned} S &= (SK, r), \\ \mathsf{Tag}_{SK,r}(m) &: c \leftarrow \mathsf{Enc}^m(r), \\ \mathsf{Ver}_{SK,r}(m,c) &: \ \textit{accept iff } \mathsf{Dec}^m(c) = r. \end{aligned}$$

Lemma 2 *If* $(\mathsf{Enc}, \mathsf{Dec})$ *is CCA-secure, then* $(\mathsf{Tag}, \mathsf{Ver})$ *is SUF-CMUA secure.*

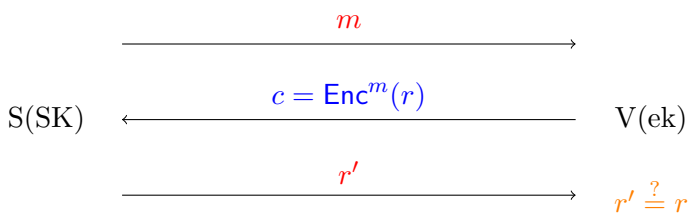
Proof: We give an intuition and leave the proof as an exercise. The intuition is that $\mathcal{A}$ must still forge a correct r even if it uses $\mathsf{Enc}(\cdot)$ for tag queries. □

Example 4: Signature Schemes

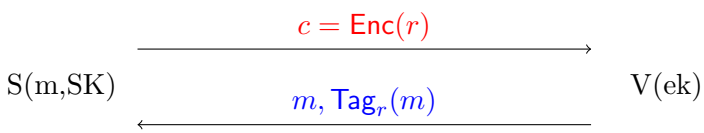
Although, without loss of generality, signatures can be deterministic, most natural sigs are probabilistic in their natural form.

Essentially, one-way signatures are equivalent to OWFs; they are, however, inefficient. In fact, every semi-efficient non-RO signature scheme is based on non-standard assumptions. Hence, one major open problem in this area concerns efficient signature schemes from DDH/CDH/Factoring. Now we show how to solve this problem using two or three rounds of interaction! In fact, more generally, we show CCA implies iCMA probabilistic signatures efficiently.

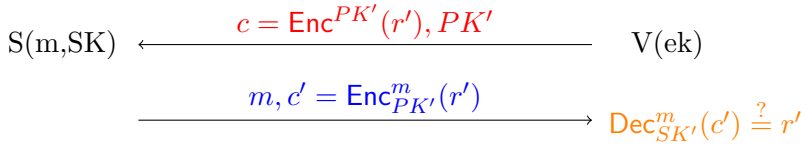
First, a 3-round construction:



Now, in the optimized 2-round version we'd like to avoid sending m as follows.



One more scheme (using MAC from CCA) is



This 2-round construction of signatures simply instantiates (with small optimizations) the MAC using CCA encryption, as explained in Example 3.