Does encryption with redundancy provide authenticity?
Jee Hea An and Mihir Bellare

Term Paper By Harkirat Singh

December 1, 2004

1 Introduction

To facilitate reliable communication between communicating parties, traditionally known as Alice & Bob, two main goals namely, *privacy* and *authenticity* needs to be satisfied. Both privacy and authenticity can be satisfied by using Encryption and MAC together, this is also called as "generic--composition". To reduce computation cost and still achieve both privacy and authenticity the most popular paradigm is called Encryption-with-redundancy, where some "redundancy" is appended to the message to be encrypted. The benefit over generic composition is that *public* redundancy avoids additional key required for MAC. Few popular approaches are OCB [3], IACBC [2], CBCC [1], & CCM [4]. The redundancy computation function could be a simple checksum, which is public and adversary can access to. On the other hand the redundancy computation function could use a shared secret key between legitimate parties and generates a cryptographic tag (like MAC). CBCC scheme is an example of checksum based redundancy function, where XOR of the message is used, we showed in the lectures that CBCC scheme is broken. Other aforesaid schemes fall into the second category and use cryptographic tag. This paper formally analysis the impact of *public* and *secret* redundancy function on authenticity of encryption-with-redundancy scheme. Base encryption scheme can be one of the three notions: IND-CPA, IND-CCA, and NM-CPA. Authors ask interesting question that given a notion of security SSS-AAA what security attributes of the redundancy code $\mathcal{RC}$ will ensure that $\mathcal{RC}$ is integrity providing w.r.t. security notion SSS-AAA.

The key findings of the paper are:

- Intuition that a strong privacy provided by encryption makes integrity strong is wrong.
- There is *no* public redundancy computation $\mathcal{RC}$ which is integrity-providing with respect to notions of security of IND-CPA, IND-CCA, and NM-CPA. This is very important result as it applies to base encryption scheme which is IND-CCA (very strong notion of privacy). *So, strong privacy does not boost integrity.*
- There is *no* secret redundancy computation $\mathcal{RC}$ which is integrity-providing with respect to base encryption scheme which meets weak notion of privacy like IND-CPA.
- If base encryption scheme meets the notion of IND-CCA or NM-CPA and secret redundancy is UF-NMA[1] (UnForgeable under No-Message Attack), means it is MAC for which it is infeasible for an adversary to output valid forgery without looking at MACs of any message, then $\mathcal{RC}$ is integrity providing with respect to IND-CCA & NM-CPA.

[1] It is very weak security requirement

- If $\mathcal{RC}$ is not UF-NMA then there exists a NM-CPA or IND-CCA encryption scheme such that the combined $\mathcal{ER}$ is not INT-CTXT secure.
- NCBC (Nested CBC) a variant of CBC where the last message block is encrypted under a key different from that used for the other message blocks. A private redundancy code which meets the property of (AXU) (Almost XOR Universal) is suffice to provide authenticity of $\mathcal{ER}$. On the other hand a public $\mathcal{RC}$ should be XOR-collision-resistant to ensure authenticity of NCBC.
- Treat each constriction of encryption-with-redundancy scheme separately and analyse it to know if the scheme meets privacy and integrity.

The next section provides some definitions to understand this paper, section 3 presents encryption-with-redundancy (both public and secret), section 4 presents details of NCBC, a variant of CBC, with redundancy.

2 Definitions

Let $\mathcal{SE} = (\mathcal{K}_e, \mathcal{E}, \mathcal{D})$ be base symmetric encryption scheme. Let $\mathcal{RC} = (\mathcal{K}_r, \mathcal{H})$ be redundancy code, where $\mathcal{K}_r$ is key generation algorithm which takes a security parameter k and returns a key K_r; $K_r \xleftarrow{\$} \mathcal{K}_r(k)$. The deterministic redundancy computation algorithm $\mathcal{H}$ takes K_r and a string $M \in \{0,1\}^*$ and returns a string τ; $\tau \leftarrow \mathcal{H}_{K_r}(M)$. The redundancy is *public* if the key K_r is public and known to adversary, if the key K_r is part of the shared key then the redundancy is *private*. The extended encryption with redundancy $\mathcal{ER} = (\mathcal{K}_c, \mathcal{K}_s, \overline{\mathcal{E}}, \overline{\mathcal{D}})$. $\mathcal{K}_c$ is randomized common key generation algorithm, $\mathcal{K}_s$ is randomized secret key generation algorithm, the difference between $\mathcal{K}_c$ & $\mathcal{K}_s$ is that unlike $\mathcal{K}_s$ which is shared between the sender and the receiver, $\mathcal{K}_c$ is shared among the two and the adversary. Note, for an extended encryption scheme with *secrete* redundancy $\mathcal{ESR}$, $\mathcal{K}_c$ returns empty string .

3 Encryption with redundancy

An extended encryption scheme with redundancy $\mathcal{ER}$ inherits the privacy of the base encryption $\mathcal{SE}$. In other words $\mathcal{RC}$ whether private or public does not play any role in determining the privacy of $\mathcal{ER}$. Let there is an adversary A which gain advantage δ against $\mathcal{ER}$, we construct an adversary B against $\mathcal{SE}$. We show it for IND-CPA case, similarly it can be shown for (IND-CCA & NM-CPA). We present pseudo-code (of Theorem 3.3) given in the paper.

```
Algorithm $B^{\mathcal{E}_{K_e}(\mathcal{LR}(\cdot,\cdot,b))}(k)$
$K_r \xleftarrow{\$} \mathcal{K}_r(k)$
If $\mathcal{ER}$ is public then $K_c \leftarrow K_r$ else $K_c \leftarrow \varepsilon$
Run A on input $(k, K_c)$; when A queries $(M_0, M_1)$ to LR oracle
$M_0' \leftarrow M_0 || \mathcal{H}_{K_r}(M_0)$; similarly $M_1'$
Y ← $LR(M_0', M_1', b)$; A ← Y;     When A outputs bit b; B outputs b
```

It is easy to see that B is effectively simulating encryption oracle for $\overline{\mathcal{E}}$. Therefore, adversary B has the same advantage as A.

3.1 Encryption with public redundancy

Suppose there exists a symmetric encryption scheme $\mathcal{SE}'$ which is IND-CCA secure (resp. IND-CPA, NM-CPA). Then there exists a symmetric encryption $\mathcal{SE}$ which is IND-CCA (resp. IND-CPA, NM-

CPA) secure, but for any redundancy code $\mathcal{RC}$, the extended encryption scheme with public redundancy $\mathcal{EPR}$ associated to $\mathcal{SE}$ and $\mathcal{RC}$ is not INT-CTXT secure. We need to show two things, first, that extended encryption scheme inherits privacy of the base scheme, second, that the extended scheme is not INT-CTXT. Theorem 4.2 given in the paper presents construction of $\mathcal{SE}$ based on $\mathcal{SE}'$, a valid forgery for $\mathcal{EPR}$, and standard reduction for privacy inheritance of $\mathcal{SE}$.

3.2 Encryption with secret redundancy

The paper defines a notion of *unforgeability under no message attack* (UF-NMA), which is the weakest form of security required by a MAC. It says that the goal of the adversary is to produce a valid message, tag pair without seeing any legitimately produced messages and tag pairs. Since, MAC and redundancy code $\mathcal{RC}$ are syntactically identical, therefore we can use same notion for them. The redundancy code $\mathcal{RC}$ is said to be *UF-NMA* secure if an adversary A has negligible advantage.

There are two results when secret redundancy is used. The negative result says that when the base encryption scheme $\mathcal{SE}'$ is IND-CPA secure, then we can construct another encryption scheme $\mathcal{SE}$ which is also IND-CPA but for any $\mathcal{RC}$, the extended encryption scheme with secret redundancy ($\mathcal{ESR}$) is not INT-CTXT secure.

The positive result says that if the base encryption scheme is IND-CCA or NM-CPA secure, the associated $\mathcal{ESR}$ with secret redundancy provides integrity if the redundancy code is UF-NMA. It is easy to see that Theorem 5.1 – 5.3 are based on standard reduction we learned in the lectures and homeworks.

4 Nested CBC (NCBC) with redundancy

NCBC is different from CBC in a way that NCBC uses different key for the last block. So, what are the security properties for the redundancy code which will provide integrity of the encryption with redundancy scheme which uses NCBC as the base encryption scheme. NCBC are constructed out of block ciphers, block-ciphers are usually modeled as "pseudorandom permutations". However, this paper uses even a stronger notion called "superpseudorandom permutations" where the adversary gets both forward and inverse permutation oracles.

4.1 NCBC with secret redundancy (SNCBC)

Let SNCBC[F,$\mathcal{RC}$] = $(\mathcal{K}_s, \overline{\mathcal{E}}, \overline{\mathcal{D}})$ be extended encryption scheme with secret redundancy associated to the encryption scheme NCBC[F] = $(\mathcal{K}_s, \mathcal{E}, \mathcal{D})$ and a redundancy code $\mathcal{RC} = (\mathcal{K}_r, H)$ as per construction 3.2 (paper). AXU-Collision property (Definition 6.2) of redundancy code says that for some $j < i$ such that $\mathcal{H}_{K_r}(X_j) \oplus y_j = \mathcal{H}_{K_r}(X_i) \oplus y_i$. AXU-Collision property of $\mathcal{RC}$ is used to determine the integrity of SNCBC. Theorem 6.5 expresses advantage of the adversary against SCNBC in INT-CTXT sense in terms of advantage against $\mathcal{RC}$ in *axu* sense and advantage against block-cipher in prp-cca sense.

It is interesting to note how Lemma 6.6 is produced, which is the information theoretic case. The lemma says that AXU-Collision security of $\mathcal{RC}$ implies security of SNCBC[F, P^l, $\mathcal{RC}$], P^l denote the family of all permutations of $l - bits$, which is the block size of NCBC. The lemma is a standard reduction proof. Let A be an adversary violating the INT-CTXT of the SNCBC scheme, using A construct an adversary B violating axu-collision security of $\mathcal{RC}$. The goal of adversary B is to output an axu-collision without knowing the key K_r. Remember, $\mathcal{RC}$ generates a block of length l and this is the last block, assume f is a function chosen from P^l, so as long as input to f is different, adversary B can simulate the output by just picking a random string from the available range of permutations.

So as long as the $n + 1^{th}$ block of the valid forgery by adversary A matches with the simulated last block, adversary B outputs 1, means AXU-Collision. But if they are different then B will not output 1, so call this BAD event, it is easy to see that this event occurs $1/(2^l - q)$, where q is the number of queries. So B will loose with this small probability.

4.2 NCBC with public redundancy (PNCBC)

A $\mathcal{RC}$ scheme which is "XOR-collision-resistant" is suffice to provide integrity. XOR-collision-resistance is slightly stronger than "collision-resistance". $\mathcal{RC} = (\mathcal{K}_r, H)$ is said to be XOR-collision-resistance (XCR) it it is hard to find strings x, x' where $x \neq x'$ such that $\mathcal{H}_{K_r}(x) \oplus H_{K_r}(x') = r$ for any committed value r and any given key K_r. HMAC is a candidate for a XCR redundancy (a keyed Hash function), SHA-1 does not yield an XCR redundancy. The paper shows the transformation of any collision-resistant function into XCR redundancy code.

References

[1] J. Jueneman, C. Meyer, and S. Matyas. Message authentication with manipulating detection codes. In *IEEE Symposium on Security and Privacy*, pages 33–54, 1984.

[2] C. Jutla. Encryption modes with almost free message integrity. In *Advances in Cryptology EUROCRYPT'01*, volume 2045, B, 2001.

[3] P. Rogaway, M. Bellare, J. Black, and T. Krovetz. Ocb: A block-cipher mode of operation for efficient authenticated encryption. In *8th Annual Conference on Computer and Communications Security*, 2001.

[4] D. Whiting, R. Housley, and N. Ferguson. Counter with cbc-mac (ccm). 2002.