Non-malleable public key encryption in BRSIM/UC

István Vajda

Abstract
We propose an extension to the BRSIM/UC library of Backes, Pfitzmann and Waidner [1] with non-malleable public key encryption. We also investigate the requirement of "full randomization" of public key encryption primitives in [1], and show that additional randomization to attain word uniqueness is theoretically not justified.

1. Introduction

Semantic security is the minimal desired notion of security for public key encryption schemes. There are several equivalent definitions of this notion of security. Intuitively, it means that anything a polynomial time adversary can compute about plaintext *m* given the ciphertext $c = E_{pk}(m)$, can also be computed without access to ciphertext *c*. The adversary learns nothing about the plaintext from the ciphertext (assumed that the adversary is restricted to probabilistic polynomial time computations).

For our purpose, the following definition will also be useful:

Definition 1: For all polynomial time computable relations R seeing $\alpha \in E(m)$ does not help one to find *m'* such that $R(m,m')$ holds. (Here $\alpha \in E(m)$ means that α is an element of the set of legal encryptions of *m*.)

This kind of security is equivalent to the indistinguishability of ciphertexts (IND), which intuitively means that any pair of ciphertexts corresponding to any pairs of different plaintexts (even selected by the adversary) are indistinguishable.

The notion of non-malleable cryptography is an extension of semantically secure cryptography:

Definition 2: For all polynomial time computable relations R seeing $\alpha \in E(m)$ does not help one to find $\beta \in E(m')$, where $\beta \neq \alpha$ such that $R(m,m')$.

Three basic modes of attacks, in order of increasing strength, are CPA, CCA-Pre (CCA1) (chosen cipher-text attacks in the preprocessing mode), CCA-Post (CCA2) (chosen cipher-text attacks in the post-processing mode); see for instance [8]. In preprocessing mode the adversary may send any polynomial number of ciphertexts to the decryption oracle before being presented with a challenge ciphertext, for which it

Technical University of Budapest E-mail: vajda@hit.bme.hu

tries to find a violation of non-malleability. In post-processing mode the adversary carries out a preprocessing attack is presented with a challenge ciphertext, and is then permitted the post-processing attack; it may query the decryption oracle with any polynomial number of ciphertexts other than the exact string, before trying to find a violation of non-malleability.

Non-malleability is a subtle concept. Note, in the knowledge of any partial information about the plaintext we can launch successful attack against malleability. In other words, if we can break IND-security, i.e. we can gain partial information about the plaintext, then based on this partial information, we can easily produce a related plaintext and a corresponding ciphertext. However, if we break NM-security, we might produce another ciphertext to a given ciphertext with related plaintext, without necessarily knowing the corresponding plaintext. This explains why we need a stronger encryption for attaining non-malleability.

It was shown in [8] that non-malleability (NM) is stronger than indistinguishability of ciphertexts: NM-AAA implies IND-AAA, $\mathrm{AAA} \in \{\mathrm{CPA}, \mathrm{CCA\text{-}Pre}, \mathrm{CCA\text{-}Post}\}$. Interestingly, NM-CCA-Post is equivalent to IND-CCA-Post. Here we cite the latter result together with its proof, because the proof is short and provides useful insight into the relationship between NM and IND security notions.

Lemma 1 [8]: Under the strongest attack of chosen ciphertext in the post-processing mode (CCA-Post) non-malleability and semantic security are equivalent.

Proof [8]:
Any violation of semantic security is clearly a violation of non-malleability: having found an *m'* related to *m* in the public-key case (equivalently, gaining partial information about *m*) it is trivial to obtain $E(m')$, where *m'* is related to *m*; just encrypt *m'* using the public key.
For the other direction if it were possible to maul an encryption $\alpha \in E(m)$ to obtain an encryption $\beta \in E(m')$ for *m'* related *m*, and where $\beta \neq \alpha$, then by feeding β to the decryption oracle one would obtain a plaintext *m'* related to *m*.
□

It follows that if under CCA-Post attack the adversary is not able to compute any partial information about the plaintext given the ciphertext then the corresponding encryption is also non-malleable.

Plaintext-awareness is also a related notion [5],[6]. A cryptosystem is plaintext-aware if it is difficult for any efficient algorithm to come up with a valid ciphertext without being aware of the corresponding plaintext. Intuitively, it also means, that the structure of the space of ciphertexts cannot be discovered by efficient algorithms. Plaintext-awareness is a very strong property. Assuming a plaintext-aware encryption, an NM-attacker is not able to fabricate a ciphertext just by beeing able to produce the corresponding plaintext, which latter effort can be foiled by assuming also semantic security. A variant of the Cramer Shoup encryption scheme was shown to be fully plaintext aware in the standard model of cryptography under the knowledge of exponent assumption (DHK) [7].

Exploration of the relationship between Dolev-Yao model and computational model of cryptography was the topic of dissertation [9]. Under (so called) weak Dolev-Yao non-malleability, the main assumptions of the Dolev-Yao model were translated into computational setting. Intuitively, weak Dolev-Yao non-malleability means that it should be hard for the computational adversary to produce a given message outside the closure of its input. It was shown that weak Dolev-Yao non-malleability is satisfied by plaintext awareness.

A stronger definition of Dolev-Yao non-malleability was also introduced in [9], which ensures security against a more adaptive adversary (actually CCA-Post adversary). In the stronger definition of non-malleability instead of the negligibility of the probability of producing a given message outside the closure of its input, similar negligibility is required for any messages outside the closure.

The formalism we follow in describing the abstract and real model for the non-malleable property of public key encryption is borrowed from [1]. It coincides with our aim to propose an extension of public key encryption with NM-property in the universally composable cryptolibrary of Backes-Pfitzmann-Waidner (BPW) [1]. The BRSIM/UC approach introduced by BPW provides a powerful tool for the security analysis of cryptographic protocols. It is a sound way of proof, where we can separate the formal and the cryptographic part by producing a symbolic model and performing the proof in this model but having the assurance that if we finally replace the symbolic cryptographic operations with real ones, the protocol which is proved to be secure in symbolic model remains secure also in the real cryptographic model. [1],[2],[3],[10],[12],[13],[14]

In the BRSIM/UC approach three different models appear for the same cryptographic primitive. Definition of the abstract (ideal) model is the starting point, which symbolic model is „dressed up" into the real model via simulation. However, when we apply a protocol certified as secure in BRSIM/UC proof framework, we have to plug in real primitives, which are secure by standard definition of security (Fig 1.) .

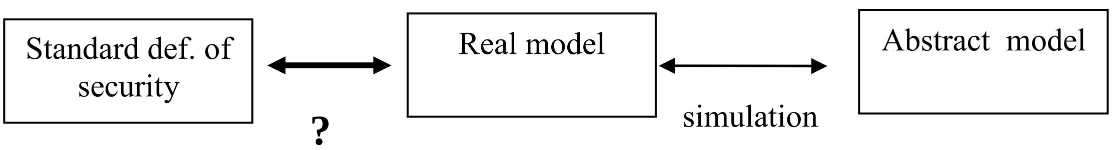

Fig.1: Security models (standard, real, abstract)

A standard definition is a notion of breaking (e.g. Semantic (IND), NM) under different attack classes in computational setting. In simulation-based approach the essential point is what we consider ideal notion of breaking. The ideal notion typically does not resemble the standard definition, even if we set aside the obvious difference coming from the abstract vs. computational theoretic terms.

Exploration of the relationship between Dolev-Yao model and computational model of cryptography was the topic in [9]. Loosely speaking, they investigated relationship directly between the left and rigth side boxes of Fig.1.

In contrast, when we try to build non-malleability into the BRSIM/UC proof system, there is an intermediate model between the Dolev-Yao type model (abstract model in Fig.1) and the computational model (standard definition in Fig.3), the real model, which is in simulation relationship with the abstract model. The real model „realizes" the abstract model (translates it into the „real world") in a cryptographically sound way. It is not obvious at all, how abstract security requirements are reflected in standard definitions. In this respect, here we investigate one important requirement of the abstract model, the „word uniqueness".

Word uniqueness requirement with respect to public key encryption means that in the database within the trusted host TH in the ideal system each new encryption is registered as a new entry of the database (i.e. modelling that they are different). For instance, if the same plaintext is encrypted with the same public key during the run, ideally we require that the corresponding ciphertexts are different. Obviously a deterministic encryption would fail to meet this requirement. Even in case of probabilistic encryption there is a positive probability that these two ciphertexts collide in the real system, i.e. word uniqueness cannot be fullfilled in an exact manner in the real system.

In [1], encryption primitives intended to implement the encryption transformation are assumed to be „fully randomized", strenghtened with additional randomization with the aim to enforce the fulfillment of the „word uniqueness" requirement of the ideal system, i.e. to make the probability of ciphertext collision event negligible. Such „full randomization" is the only, additional requirement against the applied encryption primitives, which fact also underpins the significance of our investigation.

In [1], this additional randomization is random padding of plaintexts, i.e. instead of encryption $E(m)$ we compute $E(m \| r)$, where r is binary string of genuine random bits. Pseudorandom bit sequence generators (like Blum-Micali '82) are not suitable replacements for random bit sources here [11].

All provably secure encryption algorithms are probabilistic, they need one-time genuine random bits (henceforth called inherent randomization). Therefore if we want to randomize also the space of plaintexts (henceforth called plaintext randomization) we need random bits from two reasons per encryption.

2. Our contribution

Non-malleability is a subtle notion even if we consider only public key encryption. It is not obvious at all, how to define the ideally secure model, e.g. how to introduce knowledge of relations into the abstract model.

In Section 3. we propose a model for non-malleable public key encryption for the BRSIM/UC approach of BPW. The results are interesting both theoretically and practically. Assume we have a protocol under potential malleability attack against public key encryptions within the protocol. Having proved the security of the symbolic version of a protocol we can be sure that plugging in a real non-malleable encryption primitive the real protocol remains secure. Intuition and the key ideas are introduced in this report.

Non-malleability can also be considered as an additional requirement for public key encryption, as such an encryption meets also usual security properties, for instance, semantic (IND) security is also implied. Therefore even a non-malleable public key encryption has to meet properties required in the BRSIM simulation approach for "general" public key encryption.

In Section 4 we examine one aspect of secure implementation of public key encryption of a protocol proved to be secure in the BRSIM/UC approach; the word uniqueness requirement. The significance of this investigation is given by the fact that in crypto-library [1] the only additional adjustment to real public key encryptions is that these real primitives are required to be "fully randomized". Provably secure encryption is only of theoretical interest even today because of cost reasons (cost of complexity and cost of genuine random bits per encryption).

We have an eminent interest in keeping the need for genuine random bits as low as possible. Is it practically/theoretically necessary to apply additional randomization to probabilistic public key encryption secure by corresponding standard definition of security in order to attain word uniqueness (collision avoidance) for ciphertexts?

Word uniqueness (collision avoidance) requirement does not appear (at least explicitely) in standard definitions. Is it there? We show that the standard IND-security implies word uniqueness.

3. NM secure public key encryption in BRSIM/UC

NM-attack, successful NM-attack

Assume, an honest user *B* sends a ciphertext $c = E_{pk}(m)$ to an honest user *C*, where the ciphertext suffers an NM-attack on the way to user *C* and user *C* receives a ciphertext $c' = E_{pk}(m')$, where an NM-attack is a ciphertext modification attack with the aim of producing related plaintext with a relation known by the adversary.

For simplicity, we assume that an NM-attack is the only attack type which modifies a ciphertext sent by an honest user.

Detection of the attack would be obvious, if an honest user had the coherent pair ciphertexts (*c*,*c'*) at hand (i.e. simply by recognizing that those ciphertexts are different). For generality, we exclude the existence of such detection possibility.

Participants of a protocol need to be able to uniquely identify protocol elements they process within a run. Assume protocol elements which are to be handled separately have a unique identifier (called protocol element identifier, PEI). Protocol elements per run must be unique, which means that there can not exist two different bit strings per run having the same PEI within the system.

During an NM-attack the adversary modifies a special protocol element, a ciphertext. The receiving participant decoding the fabricated ciphertext is able to decide if the result is one from the set L of legal plaintexts; when the resulted plaintext is legal, we consider the NM-attack as successful, otherwise it fails and the run will be aborted

Note, set L may depend on PEI, in general, on the role of the protocol element. Relation R from Definition 2. is defined indirectly via requirements $m' \in L$, $m' \neq m$.

In the ideal system no attack can be successful, however as an unavoidable impairment, failures of NM-attack appear simultaneously both in the ideal and the real system.

Note, the adversary does not have the possibility to check on his own if an NM-attack is successful or not, just by an indirect way. Also in this respect an NM-attack considerably differs from other usual type of attacks like attacking secret keys, or fabricating a signature.

Relation-handle, relation secrecy

In the formalism of [1], the knowledge of any information unit (e.g. a ciphertext) by a participant of the protocol is formalized by a label, so called handle, which is an element in the record modeling the information unit.

We introduce a similar indicator of the knowledge of a relation by participant *u* for the pair of plaintexts *m*, *m'* behind the pair of corresponding ciphertexts *c*, *c'*. We set a flag r_hnd_u (called relation-handle) for honest user u and a given pair of ciphertexts produced during the run (and stored in the database), which indicates that a valid relation is known by participant *u* without providing further details about the information. ($r\_hnd_u = \downarrow$ will denote that participant u is not aware of any such relation with respect to a given pair of ciphertexts.)

Relation secrecy (Definition 3. below) is the core of the definition of non-malleability property within the ideal model.

Definition 3 (Relation secrecy):
Label $r\_hnd_u \neq \downarrow$ if and only if user u has got handle to ciphertexts c and c', both, furthermore user u has got handle to the corresponding plaintexts m and m', or has got handle to the corresponding secret decoding key.

In other words, in the ideal system a participant is aware of a relation if and only if it knows (or able to get to know) the corresponding pair of plaintexts *m* and *m'* (when the participant is aware of all possible relations).

In the real system the adversary may get a valid relation-handle to a pair of ciphertexts even in case when he does not have access to the corresponding plaintexts (directly or via the corresponding secret decoding key).

Relation handle is defined for all pairs of ciphertexts produced during the examined run of the system. Therefore, whenever a new handle is given to a ciphertext (a new ciphertext is produced and its issuer gets a handle or a participant gets access to a ciphertext already existing in the system) the above condition is checked about the relation-handle between the ciphertext in concern and all already existing ciphertexts in the run.

In the ideal system no NM-attack can be successful. The success probability of the NM-attack in the real system has to be kept negligible.

Setting relation-handle for honest participants

In all commands when a participant, *u* gets a handle c^{hnd} to a ciphertext *c* algorithm *pairs2r_handle* is executed (defined precisely in subsections 3.1.and 3.2.). This algorithm outputs a relation-handle for participant *u* according to the definition of relation secrecy; if there exists an other ciphertext *c'* to which participant *u* has a handle, furthermore it has also handle also to the corresponding pair of plaintexts or it has handle to the corresponding secret decoding key.

Setting relation-handle for the adversary

In the real system the adversary may get relation-handle also by successfully running NM-attack algorithm (i.e. in addition to handles set by algorithm *pairs2r_handle*).
In case of successful NM-attack the corresponding relation-handle is set for the adversary.

Subsequently, we formalize the concept of relation-secrecy and the corresponding algorithm *pairs2r_handle* for the ideal and the real system.

3.1. The symbolic model of non-malleability

We give the definition of algorithm *pairs2r_handle* in the ideal system, which runs in trusted host *TH*. We use formalism of [1].

For better traceability of the formal definitions we cite the entry of type *enc* from [1]:

In the ideal system, if the encryption is made by TH

(ind , type := enc, arg := (pk, l), hnd_u, len) , where *ind* is the index of the entry, the argumentum (*arg*) consists of two terms, the index of the corresponding public key, *pk* and the list, *l* (encrypted content), respectively. Furthermore, *len* denotes the length of the ciphertext.

Similarly, if the ciphertext is produced within the adversarial machine, then arg:=(pk).

1. $r\_hnd_u = \downarrow$ {relation handle initially set to null}
2. $i = 0$
3. $x = size$ {the last entry with index *size* (ciphertext *c*)}
4. $y = D[ind = x].\mathrm{arg}[1]$ {index to public key}
5. $z = D[ind = x].\mathrm{arg}[2]$ {index to plaintext (if given in *arg*)}
6. **if** {checking the knowledge of the corresponding secret key or plaintext}

   $(D[ind = y-1].hnd_u \neq \downarrow \cap D[ind = y-1].type = ske) \cup D[ind = z].hnd_u \neq \downarrow$
   then
7. **do** {exhaustive search for past ciphertexts in the database (*c'*)}

8. i=i+1
9. $x = D[hnd_u = currhnd_u - i].ind$
10. **if** $D[ind = x].type = enc$ **then** {10-11. repeats 4-6.}
11. $y = D[ind = x].\mathrm{arg}[1]$; $z = D[ind = x].\mathrm{arg}[2]$
12. **if**
    $(D[ind = y-1].hnd_u \neq \downarrow \cap D[ind = y-1].type = ske) \cup D[ind = z].hnd_u \neq \downarrow$
    then

13. $r\_hnd_u = curr\_r\_hnd{+}{+}$ {if pair *c, c'* meets requirement in Definition 3, r_hnd_u is set current relation handle is increased by one}

14. end if
15. end if

16. **while** $r\_hnd_u = \downarrow \cap currhnd_u > 1$
17. end if

Fig.2. Algorithm *pairs2r_handle* in ideal system

3.2. The real model of non-malleability

Algorithm *pairs2r_handle* in the real system, which runs in protocol machine M_u , is similar to algorithm in Fig.2. The corresponding entries from [1] are the following. If the encryption is made by an honest participant *u*, then its database entry: $(\mathrm{hnd_u}$, word:=(enc,pk,c,r), type := enc, add_arg:=()) , where *r* is randomization. If a ciphertext is received from the adversary: $(\mathrm{hnd_u}$, l=(list,x_1,x_2,x_3,x_4), type := list, ()) .

$r\_hnd_u = \downarrow$
$i = 0$
$x = currhnd_u$
parse if necessary
$y = D\left[hnd_u = x\right].word[2]$ {index to public key}
$z = plain(x)^{(*)}$ {index to plaintext}
if
$y \neq \downarrow \;\cap\; D\left[hnd_u = y-1\right].type = ske \;\cup\; z \neq \downarrow$
then
do
i=i+1
$x = currhnd_u - i$
if $D\left[hnd_u = x\right].type = enc$ **then**
$y = D\left[hnd_u = x\right].word[2]; z = plain(x)$
if
($y \neq \downarrow \cap D\left[ind = y-1\right].type = ske) \cup z \neq \downarrow$
then
$r\_hnd_u = curr\_r\_hnd_u ++$
end if
end if

while $r\_hnd_u = \downarrow \;\cap\; currhnd_u > 1$
end if

Fig.3. Algorithm *pairs2r_handle* in the real system

(*) In [1], in the real system a ciphertext entry in database D_u of an honest protocol participant (machine) *u* does not contain a pointer to the plaintext, while the machine is obviously aware of the plaintext corresponding to a ciphertext, if made by itself. We use notation *plain*(*ind*), which provides the index of the plaintext entry corresponding to the ciphertext entry with index *ind*, if the ciphertext was produced by machine *u*. Algorithm *plain*() outputs $\downarrow$ if the ciphertext was not produced by machine *u* or *ind* is not a ciphertext index in database D_u .

The public key encryption primitive used in the implementation of the real model is assumed to be NM-AAA secure: under AAA class of attack the success probability of an NM-attack is negligible.

The following theorem is the main statement about the non-malleability models sketched above. According to this result, if we have proved that the symbolic version of a protocol meets the requirement set against the protocol, then plugging in the real encryption primitives the requirement will also be met except with negligible probability.

Theorem 1: The real model of non-malleability is a cryptographically sound implementation of ideal model of non-malleability.

Proof (Sketch):

Proof in [1] for public key encryption can be extended with non-malleability.

A new invariant is introduced: *relation_ secrecy*. It means that if up to time *t* relation secrecy (Definition 1) holds in the run, it will hold also at time *t+1*.

It may happen that the real system is not able to simulate the ideal system, because the adversary is successful in carrying out an NM-attack. According to the requirement of protocol element uniqueness, when different contents occur under the same PEI, the run is put into the set of *nm-error*. This is a new error event besides *collision event* and *guessing event* defined already in [1] for general public key encryption. The probability of *nm-error* is kept negligible by applying NM-secure public key encryption.
□

4. Word uniqueness in BRSIM/UC and „full randomization"

In order to enforce almost perfect implementation of „word uniqueness" requirement by public encryption primitive in the real system, i.e. to keep the probability of ciphertext collision event negligible, it is assumed in [1], that encryption primitives are strenghtened with additional randomization („fully randomized"). In this section we investigate the question of what an extent the inherent randomization of probabilistic public key encryption guaranties the desired collision avoidance as well as the question of how standard security definitions imply such collision avoidance properties.

4.1. Plaintext randomization

The additional randomization in [1] is plaintext randomization. By plaintext randomization the input (plaintext) space of the encryption algorithm is expanded.

The original plaintext is exchanged to another one selected randomly from the corresponding disjoint subset of the expanded input space. Therefore even if the same plaintext is encrypted twice or several times the plaintext is mapped into non-colliding randomized plaintexts with high probability (depending on the number of external random bits), i.e. collision avoidance for the corresponding ciphertexts can be enforced without regard to the inherent randomization of probabilistic encryption. Formally, the probability of collision between ciphertexts when the same plaintext is encrypted twice:

$$P(coll) = P(coll \mid m\_coll)P(m\_coll) + P(coll \mid \overline{m\_coll})P(\overline{m\_coll})$$

$$\begin{aligned}
&= P(coll \mid m_coll)P(m_coll) \\
&= P(coll \mid m_coll)P(m_coll) \\
&= P(coll \mid m_coll) \cdot \frac{1}{2^{rand_len}} \\
&\leq \frac{1}{2^{rand_len}}
\end{aligned}$$

where *coll* and *m_coll* defines the event of collision of ciphertexts and collision of randomized plaintexts, respectively, furthermore *rand_len* denotes the number of external random bits (note, $P(coll \mid \overline{m\_coll}) = 0$). Decrease of collision probability is exponential in the length of the random string.

Generalization to more than two (in general, to polynomial number) of colliding plaintexts is straightforward.

4.2. Inherent randomization

In probabilistic encryption by the inherent randomization the output space of the encryption mapping is expanded. Disjoint subsets of ciphertexts in the output space are assigned to different plaintexts for unique decoding. When a plaintext is encrypted a block of random bits is generated and, accordingly, a random element is chosen from the corresponding subset of ciphertexts in the expanded output space.

We assume that the log-size of these subsets is equal to the number w of random bits. When different users of the system, or the same user in concurrent runs encrypt the same plaintext using the same public key, it may happen that the same random bits are selected, which leads to collision between the corresponding ciphertexts. Let M denote the number of users/run encrypting a given plaintext m. Let $W(k) = 2^{w(k)}$ be called the expansion factor, which depends on security parameter k.

Lemma 2: Super-polynomial grow of expansion factor W(k) is necessary and sufficient to attain negligible collision probability.

Proof:
An analogue question is the following: what is the probability P_{coll} that when placing randomly M balls into W urns there will be at least one urn with two or more balls ($W > M$)? Note, it is a version of the birthday paradox:

$$1 - P_{coll} = \frac{M!\binom{W}{M}}{W^M} = \prod_{j=1}^{M-1}\left(1 - \frac{j}{W}\right) \qquad (1)$$

From (1) it can be seen that by increasing ratio W/M the probability of collision event can be decreased to an arbitrary small value. On the other side, by using upper bound $1 - P_{coll} < e^{-M(M-1)/(2W)}$, i.e.

$$P_{coll} > 1 - e^{-M(M-1)/(2W)}$$

it follows, for instance, that if $W/M \le M-1$, then $P_{coll} > 0.393$.

It is also obvious from the urn model, that if W/M decreases to one, then P_{coll} increases to $1 - 1/W^W$.

From this short analysis it follows that, in principle, depending on the ratio W/M the probability of collision can take "any" value from zero to one.

Taking into account the real meaning and the expected range of quantities W and M refines the answer. Assuming $w \sim O(k)$, $M \sim p(k)$ for some polynomial *p*, where *k* is the security parameter, the collision probability P_{coll} becomes asymptotically negligible.

(This assumption is met, for example, for public key encryptions: Goldwasser-Micali with RSA, Benaloh, ElGamal, Cramer-Shoup.)

Note, that instead of an exponential grow of $W(k)$, a super-polynomial grow also ensures the wanted negligibility of probability P_{coll}.
□

We can step forward by approaching the problem from a different side. Requirement of word uniqueness (collision freeness) does not appear explicitely in standard definitions. However, implicite, it is there, because:

Theorem 2: The standard (IND) security definition implies the requirement of word uniqueness.

Proof:
By Lemma 2 a super-polynomial grow of the expansion factor W(k) is necessary and sufficient to attain negligible collision probability. Here we prove if this grow is only polynomial, the fulfillment of the standard (plaintext indistinguishability) requirement fails: based on collision detection the plaintext pairs of the IND notion become distinguishable non-negligibly. The details follow:

Recall, different plaintexts are encrypted into disjoint sets of ciphertext, i.e. ciphertext corresponding to different plaintexts cannot collide. Now we define the IND breaking algorithm based on collision detection:

The adversary chooses a pair of plaintexts and forwards them to the challenge oracle. The oracle randomly selects one of them and returns it encrypted to the adversary. The adversary encrypts both plaintexts and decides on the ciphertext (and, accordingly, the corresponding plaintext) which collides with the ciphertext presented by the oracle. If no collision occurs, the adversary decides by flipping a coin.

Now we show that this attack algorithm provides successful decision with non-negligible probability (over 1/2):

By affording polynomial many challenges, the adversary could decide with non-negligible advantage: the adversary sees polynomial many pair of ciphertexts, where the first term is produced by the oracle and it is an encryption of the plaintext chosen by the oracle and fixed afterwards, similarly the second term of the pair is produced by the adversary and it is an encryption of the plaintext chosen by the adversary and fixed afterwards. If the adversary could not decide successfully with non-negligible probability using only one such pair, he could not do the same relying on polynomial many pairs.
□

It follows that external randomization is not justified theoretically: if an encryption algorithm meets the requirement of semantic (IND) security (under the considered attack class) word uniqueness is implied. In other words, the inherent randomization ensures the desired collision avoidance. Note, NM-security implies IND-security.

5. Conlusion

We have introduced the non-malleability property of public key encryption into the BRSIM/UC proof system of BPW. It is theoretically interesting to formalize an ideal model for non-malleable encryption, which can be realized in cryptographically sound way. The practical benefit is the extension of the set of protocols the security of which can be analyzed and proved in this proof system.

Genuine random bits are ideal elements provided by a trusted third party (atomic random oracle). In this respect, implicitly, all cryptographic algorithms assume a TTP, which is a costly source. We have examined one aspect of randomization in the BRSIM/UC proof system of BPW with respect to public key encryption. It is important to keep the amount of random bits consumed by cryptographic algorithms as low as required by necessary conditions.

References

[1] M. Backes, B. Pfitzmann, and M. Waidner. A universally composable cryptographic library. *IACR Cryptology ePrint Archive*, Report 2003/015, http://eprint.iacr.org/, January 2003.

[2] M. Backes and C. Jacobi. Cryptographically sound and machine-assisted verification of security protocols. In *Proc. 20th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 2607 of *Lecture Notes in Computer Science*, pages 675–686. Springer, 2003.

[4] M. Backes and B. Pfitzmann. A General Composition Theorem for Secure Reactive Systems. *Theory of Cryptograpy Conference (TCC 2004)*, LNCS 2951, pp. 336-354, 2004.

[5] M. Bellare and P. Rogaway. Optimal Asymmetric Encryption -- How to encrypt with RSA. *Extended abstract in Advances in Cryptology - Eurocrypt '94* Proceedings, Lecture Notes in Computer Science Vol. 950, A. De Santis ed, Springer-Verlag, 1995.

[6] M. Bellare and A. Palacio. Towards Plaintext-Aware Public-Key Encryption without Random Oracles. *In Advances in Cryptology -- ASIACRYPT 2004*, Lecture Notes in Computer Science Vol. 3329, Springer-Verlag, 2004.

[7] A. W. Dent *The Cramer-Shoup Encryption Scheme Is Plaintext Aware in the Standard Model*. *In Advances in Cryptology -- EUROCRYPT 2006*, Lecture Notes in Computer Science Vol. 4004, Springer-Verlag, 2006.

[8] D. Dolev, C. Dwork and M. Naor. Nonmalleable cryptography. *SIAM J. Comput., Vol. 30, No. 2*, pp. 391-437.

[9] J. Herzog. Computational Soundness for Standard Assumptions of Formal Cryptography, *PhD Dissertation, MIT, 2004.*

[10] B. Pfitzmann and M. Waidner. Composition and integrity preservation of secure reactive systems. In *Proc. 7th ACM CCS*, pages 245–254, 2000.

[11] R.L. Rivest and A.L. Sherman. Randomized encryption techniques, *Advances in Cryptology: Proceedings of Crypto '82*

[12] I.Vajda. Cryptographically Sound Security Proof for On-Demand Source Routing Protocol EndairA. *Cryptology ePrint Archive Report 2011/103*. http://eprint.iacr.org/2011/103.pdf

[13] I.Vajda. Framework for Security Proofs for Reactive Routing Protocols in Multi-Hop Wireless Networks. *Cryptology ePrint Archive Report 2011/237*. http://eprint.iacr.org/2011/237.pdf

[14] I.Vajda. New look at impossibility result on Dolev-Yao models with hashes. *Cryptology ePrint Archive Report 2011/335*. http://eprint.iacr.org/2011/335.pdf