2009 – 2011 Cooperative Research Project: "Security Evaluation and Design of Components and Cryptographic Primitives for RFID and Sensor Networks"

The First Spring Working Meeting

India, Kolkata,

Indian Statistical Institute

April 13-15, 2009

Moderator: Miodrag Mihaljevic

Spring Working Meeting

- Venue: Indian Statistical Institute, Kolkata
- Time: April 13-15, 2009

Participants:

- Two Members of Japan-team
- Four Members of Indian-team

Agenda of the Meeting

- **Main Goal**: “Technical Discussions on Future Joint Research Activities and Regarding the 2009 Visits of India-team Members to Tokyo”
- **Lectures** on Certain Topic Candidates for joint Research Activities Related to Crypto&Coding
- **Short Technical Talks**
- **Round-table Discussions**
- Establishing **Program of the Visit and Research Directions** for Indian-team Members Visits to Japan

Lecture I

Towards Low Complexity and Highly Secure Cryptographic Primitives Which Involve Pure Randomness and Dedicated Coding

On Algebraic Representation and Security Evaluation of Certain Stream Ciphers which Involve Randomness

Miodrag Mihaljevic

Research Center for Information Security,
National Institute of Advanced Industrial Science and Technology

Spring Working Meeting of Japan-India Project

Kolkata, April 13, 2009

Roadmap

- Introduction
- Certain Stream Ciphers Based on Randomness
- Dedicated Wire-Tap Channel Coding
- Algebraic Representation of Encryption
- Security Evaluation
- Concluding Remarks

I. Introduction

previous art
and
motivation for the work

Recent Novel Designs

- Different variants of **Stream Ciphers** based on dedicated encoding and pure randomness:
- Generic Framework
- Stream Ciphers based on **Channel with Insertion and Complementing**
- Stream Ciphers based on **Wire-Tap Channel Coding**
- **Low Complexity Authentication Protocols** based on dummy and effective bits
- Above novel designs originate from the HB class of authentication protocols

Very Recent References (1)

- [1] M. Mihaljevic and H. Imai, **"An Approach for Stream Ciphers Design Based on Joint Computing over Random and Secret Data"**, *COMPUTING*, accepted for publication, 2009. (Impact Factor: 0.949)
- [2] M. Mihaljevic, **"A Framework for Stream Ciphers Based on Pseudorandomness, Randomness and Error-Correcting Coding**", in *Enhancing Crypto-Primitives with Techniques from Coding Theory*, Editors B. Preneel and S. Dodunekov, Vol. in the Series *Information and Communication Security*, IOS Press, Amsterdam, 23 pages, to appear 2009.
- [3] M. Mihaljevic and H. Imai, **"A Stream Cipher Design Based on Embedding of Random Bits"**, *IEEE 2008 Int. Symp. on Inform. Theory and its Appl. - ISITA2008*, Auckland, New Zealand, Dec. 7-10, 2008, Proceedings, pp. 1497-1502. (ISBN: 978-1-4244-2069-8; copyright2008 IEEE)

Very Recent References (2)

- [4] M. Mihaljevic and H. Imai, **"A Stream Ciphering Approach Based on the Wire-Tap Channel Coding"**, *8th Central European Conference on Cryptography - CECC 2008*, Graz, Austria, July 2-4, 2008, Conference Records, pp. 16-18.
- [5] M. Mihaljevic, **"A Framework for Stream Ciphers Based on Pseudorandomness, Randomness and Error-Correcting Coding"**, Invited Talk at NATO Advanced Research Workshop "*Enhancing Crypto-Primitives with Techniques from Coding Theory*", 6 - 9 October 2008.
- [6] M. Mihaljevic, H. Watanabe and H. Imai, **"A Cellular Automata Based HB#-like Low Complexity Authentication Technique"**, *IEEE 2008 Int. Symp. on Inform. Theory and its Appl. - ISITA2008*, Auckland, New Zealand, Dec. 7-10, 2008, Proceedings, pp. 1355-1360. (ISBN: 978-1-4244-2069-8; copyright2008 IEEE)

Some Earlier Results on Crypto&Coding

(there is a number of other results achieved in the period 2005-2008)

- [7] M. Mihaljevic, M. Fossorier and H. Imai, **"Key Management with Minimized Secret Storage Employing an Erasure Channel Approach"**, *IEEE Communications Letters*, vol. 9, pp. 741-743, Aug. 2005. (Impact Factor: 0.922)
- [8] M. Fossorier, M. Mihaljevic, H. Imai, Y. Cui and K. Matsuura, **"An Algorithm for Solving the LPN Problem and its Application to Security Evaluation of the HB Protocols for RFID Authentication",** *Lecture Notes in Computer Science*, vol. 4329, pp. 48-62, Dec. 2006. (Impact Factor: ~ 0.5)
- [9] M. Mihaljevic, **"Generic framework for secure Yuen 2000 quantum-encryption employing the wire-tap channel approach",** *Physical Review A*, vol. 75, no. 5, pp. 052334-1-5, May 2007. (Impact Factor: ~ 3.0)
- [10] M. Fossorier, M. Mihaljevic and H. Imai, **"Modeling Block Encoding Approaches for Fast Correlation Attack",** *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4728-4737, Dec. 2007. (Impact Factor: 2.183)

Some Previous Results on Randomized Encryption

- [11] R. Rivest and T. Sherman, **"Randomized Encryption Techniques",** *Advances in Cryptology: Proceedings of CRYPTO '82*, Plemum, New Yourk, pp. 145-163, 1983.
- [12] N.J.A. Sloane, **"Error-correcting codes and Cryptography",** *Cryptologia*, vol. 6, pp. 128-153, 1982.
- [13] O. Kara and I. Erguler, **"A New Approach to Keystream Based Cryptosystems",** *SASC 2008*, Workshop Record, pp. 205-221, Feb.

Motivation for the Work

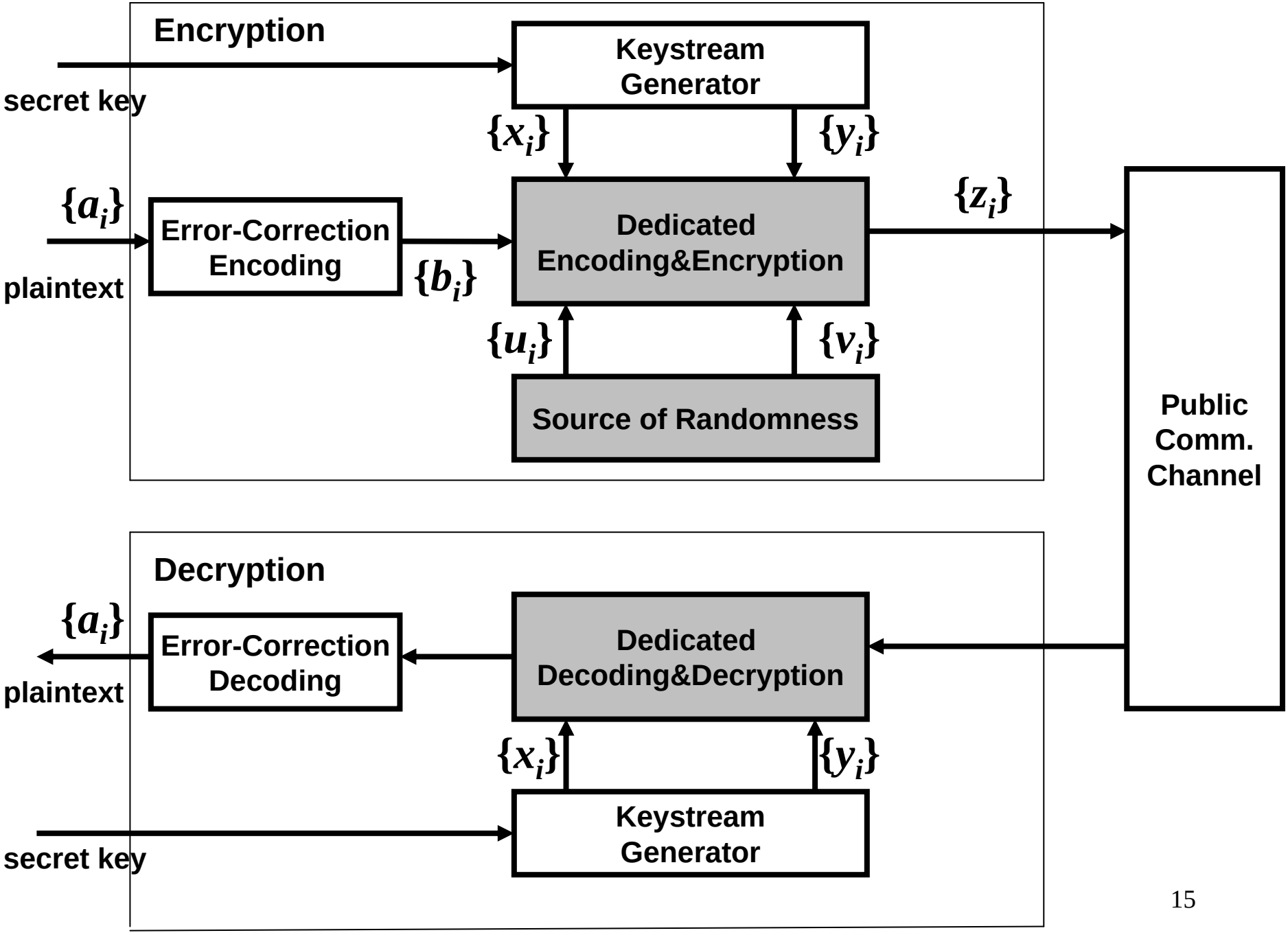
- **Establishing algebraic model** of certain stream ciphers which involve randomness and dedicated homophonic encoding.
- **Security evaluation of the established algebraic model**.

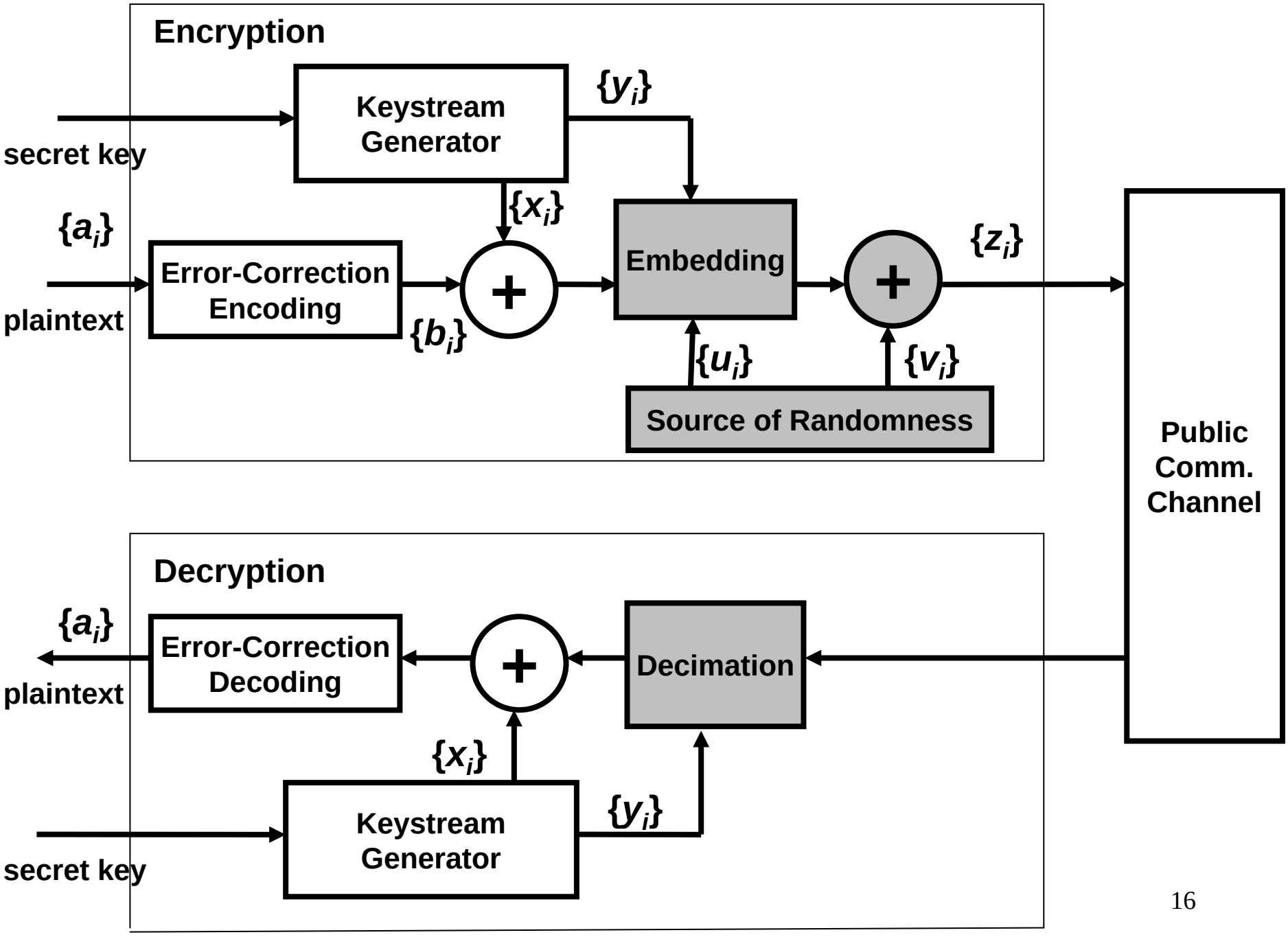
II. Certain Stream Ciphers Based on Randomness

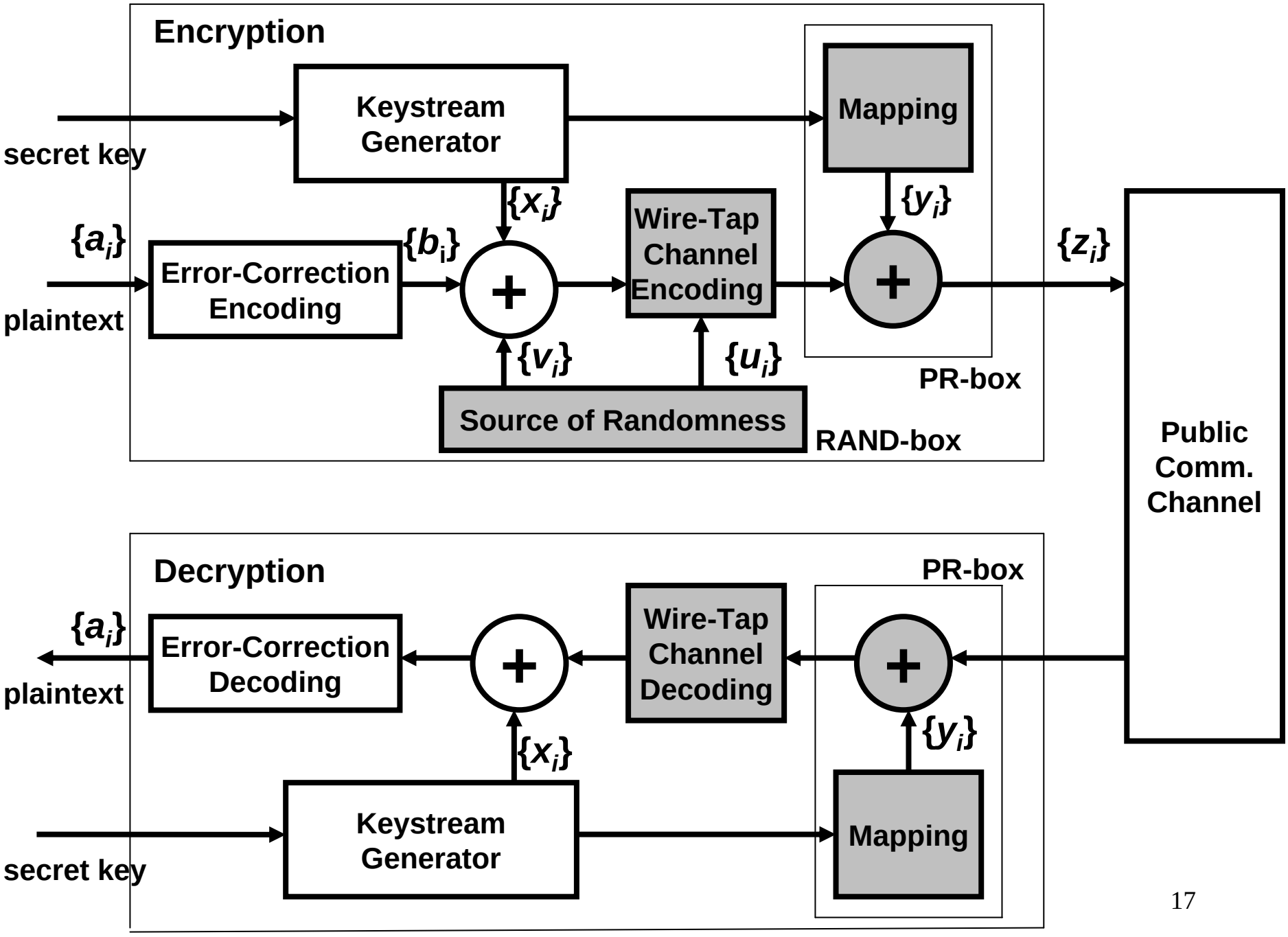
-a generic scheme

-a scheme based on simple embedding: Stream Cipher I

-a wire-tap channel coding based scheme: Stream Cipher II


Encryption
secret key
Keystream Generator
$\{x_i\}$
$\{y_i\}$
$\{a_i\}$
plaintext
Error-Correction Encoding
$\{b_i\}$
Dedicated Encoding&Encryption
$\{z_i\}$
$\{u_i\}$
$\{v_i\}$
Source of Randomness
Public Comm. Channel
Decryption
$\{a_i\}$
plaintext
Error-Correction Decoding
Dedicated Decoding&Decryption
$\{x_i\}$
$\{y_i\}$
Keystream Generator
secret key


Encryption
Keystream Generator
$\{y_i\}$
secret key
$\{x_i\}$
$\{a_i\}$
Error-Correction Encoding
Embedding
$\{z_i\}$
plaintext
$\{b_i\}$
$\{u_i\}$
$\{v_i\}$
Source of Randomness
Public Comm. Channel
Decryption
$\{a_i\}$
Error-Correction Decoding
Decimation
plaintext
$\{x_i\}$
Keystream Generator
$\{y_i\}$
secret key


Encryption
Keystream Generator
secret key
Mapping
{x_i}
{y_i}
{a_i}
{b_i}
Error-Correction Encoding
plaintext
Wire-Tap Channel Encoding
{z_i}
{v_i}
{u_i}
PR-box
Source of Randomness
RAND-box
Public Comm. Channel
Decryption
PR-box
{a_i}
Error-Correction Decoding
plaintext
Wire-Tap Channel Decoding
{x_i}
{y_i}
Keystream Generator
secret key
Mapping

III. Dedicated Wire-Tap Channel Coding

Coding Method and Selection of the Code

Coding Method (1)

We consider a generic approach for wire-tap channel coding as follows.

- To transmit m-bit message we first select a (n, k) code C such that $m \leq n - k$.

- Out of the 2^{n-k} cosets of C, we choose 2^m cosets and let each message correspond to a chosen coset.

- The selection of the cosets is done in a linear fashion as follows:

(a) Suppose $\mathbf{G}$ is a generator matrix for C with rows $\mathbf{g}_1$, $\mathbf{g}_2$, ..., and $\mathbf{g}_k$.

(b) We select m linearly independent vectors $\mathbf{h}_1$, $\mathbf{h}_2$, ..., $\mathbf{h}_m$, from $\{0,1\}^n \backslash C$.

(c) The coset corresponding to a m-bit message $\mathbf{s} = [s_1, s_2, ..., s_m]$ is determined as follows:

$$\mathbf{s} \rightarrow s_1\mathbf{h}_1 \oplus s_2\mathbf{h}_2 \oplus ... \oplus s_m\mathbf{h}_m \oplus C. \qquad (1)$$

Coding Method (2)

The above correspondence is deterministic, but the encoding has a random component in the selection of the employed codeword. The transmitted word $\mathbf{c}$ is specified as follows:

$$\mathbf{c} = s_1\mathbf{h}_1 \oplus s_2\mathbf{h}_2 \oplus ... \oplus s_m\mathbf{h}_m \oplus u_1\mathbf{g}_1 \oplus u_2\mathbf{g}_2 \oplus ... \oplus u_k\mathbf{g}_k \quad (1)$$

where $\mathbf{u} = [u_1, u_2, ..., u_k]$ is an uniformly random k-bit vector and in a particular case $k = n - m$.

The overall encoding operation can be described as the following. Let $\mathbf{G}^*$ be the $m \times n$ matrix with rows $\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_m$. Then

$$\mathbf{c} = [\mathbf{su}] \begin{bmatrix} \mathbf{G}^* \\ \mathbf{G} \end{bmatrix} \quad (2)$$

Selection of the Code

For an arbitrary m-bit message $\mathbf{S} = \mathbf{s}$, the transmitted word belongs to $\mathbf{sG}^* \oplus C$. Since the cosets of C cover the entire space $\{0,1\}^n$, the attacker receives vector $\mathbf{Z}$ which belongs to some coset of C for example $\mathbf{rG}^* \oplus C$. If $\mathbf{e}$ denotes the error vector introduced by the wire-tapper's BSC(p), we have for $1 \leq i \leq 2^k$:

$$\text{Prob}\{Z \in \mathbf{rG}^* \oplus C\} = \text{Prob}\{\mathbf{e} \in (\mathbf{u} \oplus \mathbf{s})\mathbf{G}^* \oplus C\} = \text{Prob}\{\mathbf{e} \in \mathbf{w} \oplus C\}\ , \tag{1}$$

for some n-tuple $\mathbf{w}$. Accordingly, the following criterion for selecting the code C provides security of the message: Select C such that for any n-tuple $\mathbf{w}$, the following is valid:

$$\text{Prob}\{\mathbf{e} \in \mathbf{w} \oplus C\} \to 2^{-k}, \ \ as \ \ n \to \infty. \tag{2}$$

The above condition in conjunction with (10) implies that for an attacker it is equally likely to find $\mathbf{Z}$ in any coset of C given any message $\mathbf{S}$. Note that, assuming all $\mathbf{S} = \mathbf{s}$ are equally likely *a priori*, $\text{Prob}\{\mathbf{Z} \in \mathbf{rG}^* \oplus C\}$ is independent of $\mathbf{r}$: Hence,

$$\text{Prob}\{\mathbf{S} = \mathbf{s} | \mathbf{Z} \in \mathbf{rG}^* \oplus C\} \to 2^{-k}\ , \tag{3}$$

implies the security.

IV. Algebraic Representation of the Encryption

Stream Cipher I: Encryption Algorithm

1. Encode $\mathbf{a}_l \in \{0,1\}^\ell$ into the codeword $C(\mathbf{a}_l) \in \{0,1\}^m$ employing the selected ECC suitable for a binary symmetric channel with the crossover probability η.

2. Employing the output vector $\mathbf{x}_t \in \{0,1\}^m$ from the keystream generator compute $C(\mathbf{a}_t) \oplus \mathbf{x}_t$, where $\oplus$ denotes bit-by-bit $mod 2$ addition.

3. Generate by the RAND-box a random vector $\mathbf{u}_t \leftarrow \{0,1\}^{n-m}$ and perform psudorandom embedding (controlled by the keystream generator) of the bits from the vectors $C(\mathbf{a}_t) \oplus \mathbf{x}_t$ and $\mathbf{u}_t$ as follows: $(C(\mathbf{a}_t) \oplus \mathbf{x}_t || \mathbf{u}_t)\mathbf{P}_t$, where $\mathbf{P}_t$ is a permutation matrix (selected according to the keystream generator output $\mathbf{y}_t$) which corresponds to the considered embedding and $||$ denotes concatenation.

4. Generate by the RAND-box a random $\mathbf{v}_t \leftarrow \mathrm{Ber}_{n,\eta}$ and generate the ciphertext vector as follows:

$$\mathbf{z}_t = (C(\mathbf{a}_t) \oplus \mathbf{x}_t || \mathbf{u}_t)\mathbf{P}_t \oplus \mathbf{v}_t \ . \qquad (1)$$

Stream Cipher I: Decryption Algorithm

1. Perform decimation of $\mathbf{z}_t$ corresponding to the embedding performed in the encryption step 3 as follows:

$$\mathbf{z}_t\mathbf{P}_t^{-1} = (C(\mathbf{a}_t) \oplus \mathbf{x}_t||\mathbf{u}_t) \oplus \mathbf{v}_t\mathbf{P}_t^{-1}) \ , \qquad (1)$$

$$tcat_m(\mathbf{z}_t\mathbf{P}_t^{-1}) = C(\mathbf{a}_t) \oplus \mathbf{x}_t \oplus tcat_m(\mathbf{v}_t\mathbf{P}_t^{-1}) \ , \ (2)$$

where $\mathbf{P}_t^{-1}$ denotes the inverse permutation of $\mathbf{P}_t$ and $tcat_m(\cdot)$ denotes truncating of the argument to the first m bits.

2. Employing the output vector $\mathbf{x}_t \in \{0,1\}^m$ from the keystream generator compute

$$tcat_m(\mathbf{z}_t\mathbf{P}_t^{-1}) \oplus \mathbf{x}_t = C(\mathbf{a}_t) \oplus tcat_m(\mathbf{v}_t\mathbf{P}_t^{-1}) \ . \ (3)$$

3. Perform decoding $C^{-1}(\cdot)$ of the employed ECC and recover $\mathbf{a}_t$ according to the following:

$$\mathbf{a}_t = C^{-1}(C(\mathbf{a}_t) \oplus tcat_m(\mathbf{v}_t\mathbf{P}_t^{-1})) \ . \qquad (4)$$

Stream Cipher II: Encryption Algorithm

1. Encode $\mathbf{a}_t \in \{0,1\}^\ell$ into the codeword $\mathbf{b} = C(\mathbf{a}_t) \in \{0,1\}^m$ employing the selected ECC suitable for a binary symmetric channel with the crossover probability η.

2. Employing the output vectors $\mathbf{x}_t \in \{0,1\}^m$ from the keystream generator and $\mathbf{v}_t$ from RAND-box compute $\mathbf{b}_t \oplus \mathbf{x}_t \oplus \mathbf{v}_t$, where $\oplus$ denotes bit-by-bit $mod2$ addition, and $\mathbf{v}_t \leftarrow \mathrm{Ber}_{m,\eta}$.

3. Generate by the RAND-box a random vector $\mathbf{u}_t \leftarrow \{0,1\}^{n-m}$ and perform the selected wire-tap channel coding $C_W(\cdot)$ which provides the codeword $C_W(\mathbf{b}_t \oplus \mathbf{x}_t \oplus \mathbf{v}_t, \mathbf{u}_t)$ as an n-dimensional binary vector.

4. Employ $\mathbf{y}_t \in \{0,1\}^n$ and generate the ciphertext vector as follows:

$$\mathbf{z}_t = C_W(\mathbf{b}_t \oplus \mathbf{x}_t \oplus \mathbf{v}_t, \mathbf{u}_t) \oplus \mathbf{y}_t \ . \qquad (1)$$

Stream Cipher II: Decryption Algorithm

1. Calculate:

$$\mathbf{z}_t \oplus \mathbf{y}_t = C_W(\mathbf{b}_t \oplus \mathbf{x}_t \oplus \mathbf{v}_t, \mathbf{u}_t) \; . \qquad (1)$$

2. Perform decoding $C_W^{-1}(\cdot)$, corresponding to the employed wire-tap channel encoding, as follows:

$$C_W^{-1}(\mathbf{z}_t \oplus \mathbf{y}_t) = \mathbf{b}_t \oplus \mathbf{x}_t \oplus \mathbf{v}_t \; . \qquad (2)$$

3. Employing the output vector $\mathbf{x}_t \in \{0,1\}^m$ from the keystream generator compute

$$C_W^{-1}(\mathbf{z}_t \oplus \mathbf{y}_t) \oplus \mathbf{x_t} = C(\mathbf{a}_t) \oplus \mathbf{v}_t \; . \qquad (3)$$

4. Perform decoding $C^{-1}(\cdot)$ of the employed ECC and recover $\mathbf{a}_t$ according to the following:

$$\mathbf{a}_t = C^{-1}(C_W^{-1}(\mathbf{z}_t \oplus \mathbf{y}_t) \oplus \mathbf{x_t}) \; . \qquad (4)$$

Unified Algebraic Model

for Stream Ciphers I & II

For the Stream Cipher I, the following algebraic model can be shown:

$$\mathbf{z} = (\bigoplus_{i=1}^{m}(b_i \oplus x_i)\mathbf{p}_i) \oplus (\bigoplus_{i=1}^{n-m} u_i \mathbf{p}_{m+i}) \oplus \mathbf{v} \ ,$$

$$= (\bigoplus_{i=1}^{m} b_i \mathbf{p}_i) \oplus \phi(\mathbf{x}, \mathbf{u}, \mathbf{v}, \mathbf{P}) \qquad (1)$$

where $\phi(\mathbf{x}, \mathbf{u}, \mathbf{v}, \mathbf{P}) = (\bigoplus_{i=1}^{m} x_i \mathbf{p}_i) \oplus (\bigoplus_{i=1}^{n-m} u_i \mathbf{p}_{m+i} \oplus \mathbf{v})$, $\mathbf{b} = C(\mathbf{a})$, and $\mathbf{P}$ is a function of $\mathbf{y}$, i.e. $\mathbf{k}$.

For the Stream Cipher II, assuming that the linear wire-tap channel coding $C_W(\cdot)$ is employed, after the wire-tap channel encoder, we have the following:

$$\mathbf{z} = C_W(\mathbf{b} \oplus \mathbf{x} \oplus \mathbf{v}, \mathbf{u}) \oplus \mathbf{y} = (\bigoplus_{i=1}^{m} (b_i \oplus x_i \oplus v_i)\mathbf{h}_i) \oplus (\bigoplus_{i=1}^{n-m} u_i \mathbf{g}_i) \oplus \mathbf{y} \quad (1)$$

$$= (\bigoplus_{i=1}^{m} b_i \mathbf{h}_i) \oplus \phi(\mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{v}, \mathbf{G}, \mathbf{G}^*) \; . \quad (2)$$

where

$$\phi(\mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{v}, \mathbf{G}, \mathbf{G}^*) = (\bigoplus_{i=1}^{m} x_i \mathbf{h}_i \oplus \mathbf{y}) \oplus (\bigoplus_{i=1}^{n-m} u_i \mathbf{g}_i) \oplus (\bigoplus_{i=1}^{m} v_i \mathbf{h}_i). \quad (3)$$

Proposition 1. In the both Stream Ciphers I and II, the ciphertext $\{z_t\}$ can be represented as follows:

$$\mathbf{z} = f_i(\mathbf{a}) \oplus \phi_i(\mathbf{k}, \mathbf{r}) \ , \ \ i = I, II \ , \qquad (1)$$

where for the Stream Cipher I

$$f_I(\cdot) = \bigoplus_{i=1}^{m} b_i \mathbf{p}_i \ , \ \ \phi_I(\cdot) = (\bigoplus_{i=1}^{m} x_i \mathbf{p}_i) \oplus (\bigoplus_{i=1}^{n-m} u_i \mathbf{p}_{m+i}) \oplus \mathbf{v} \ , \qquad (2)$$

and for the Stream Cipher II

$$f_{II}(\cdot) = \bigoplus_{i=1}^{m} b_i \mathbf{h}_i \ , \ \ \phi_{II}(\cdot) = (\bigoplus_{i=1}^{m} x_i \mathbf{h}_i \oplus \mathbf{y}) \oplus (\bigoplus_{i=1}^{n-m} u_i \mathbf{g}_i \oplus \bigoplus_{i=1}^{m} v_i \mathbf{h}_i), \qquad (3)$$

and where $\mathbf{b} = C(\mathbf{a})$, $\mathbf{k}$ denotes secret key, $\mathbf{r}$ is a random vector, and $\mathbf{P} = [\mathbf{p}_i]_{i=1}^{n}$ is a permutation $n \times n$ matrix, and each $\mathbf{p}_i$ is a binary n-dimensional vector with only one non-zero element.

V. Security Evaluation

LPN Problem Based Security

Indistinguishability

One of the security goals is the indistinguishability (IND): IND deals with the secrecy provided by the scheme in the following sense: An adversary must be unable to distinguish the encryption of two (chosen) plaintexts. For the IND considerations we assume the following traditional approach. An adversary is considered as a pair of algorithms $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and they operate through two phases as follows.

Phase I. $\mathcal{A}_1$ is employed during the first phase and at the end of this phase, $\mathcal{A}_1$ outputs a pair of plaintexts $(\mathbf{x}_1, \mathbf{x}_2)$.

Phase II. One of the given plaintexts is selected with probability equal 1/2, then encrypted, and the obtained ciphertext is delivered to $\mathcal{A}_2$ - this represents $\mathcal{A}$'s challenge. The success of $\mathcal{A}$ is determined according to correctness of decision whether $\mathbf{x}_1$ or $\mathbf{x}_2$ was encrypted.

LPN Problem

Informally, Learning from Parity with Noise (LPN) problem can be described as learning an unknown k-bit vector $\mathbf{s}$ given noisy versions of its scalar product $\mathbf{a} \cdot \mathbf{s}$ with randomly selected vectors $\mathbf{a}$.

In a formal manner, the LPN problem is the problem of retrieving $\mathbf{s}$ given access to the oracle $\Pi_{s,\eta}$. For a fixed value of k, we will say that an algorithm $\mathcal{A}(T, q, \delta)$-solves the LPN problem with noise parameter η if $\mathcal{A}$ runs in time at most T, makes at most q oracle queries, and

$$\Pr\left[\mathbf{s} \leftarrow \{0,1\}^k \ : \ \mathcal{A}^{\Pi_{s,\eta}}(1^k) = s\right] \geq \delta$$

By saying that the LPN problem is hard, we mean that any efficient adversary solves it with only negligible probability. There is a significant amount of literature dealing with the hardness of the LPN problem. It is closely related to the problem of decoding a random linear code and it is NP-hard.

LPN Problem and the Indistinguishability

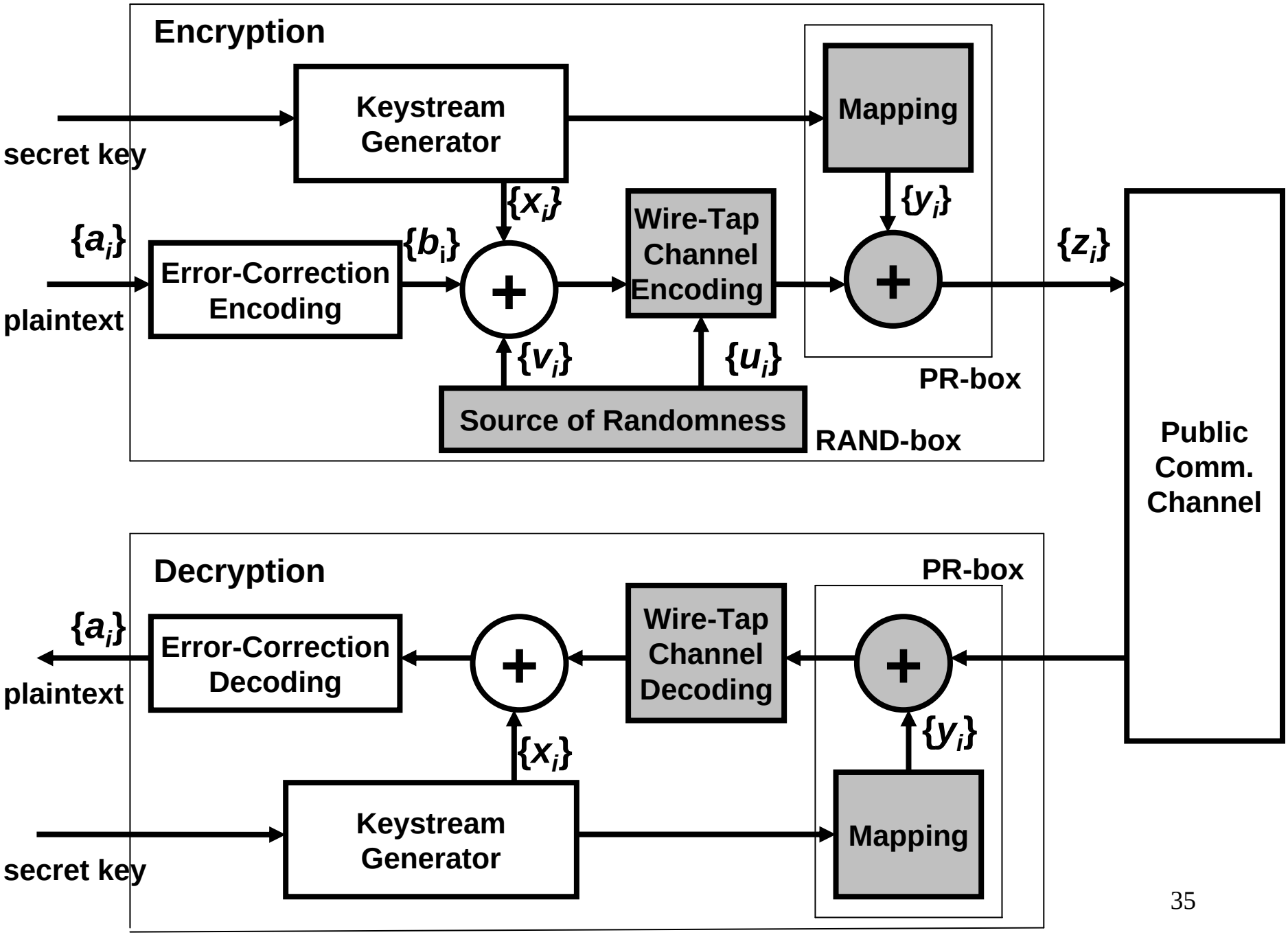
The following lemma states that the hardness of the LPN problem implies that the two oracles $\mathcal{U}_{k+1}$ and $\Pi_{s,\eta}$ are indistinguishable.

Lemma 1, [Katz, CRYPTO&Coding2007]. Assume there exists an algorithm $\mathcal{M}$ making q oracle queries, running in time T, and such that

$$\left|\Pr\left[\mathrm{s} \leftarrow \{0,1\}^k \ : \ \mathcal{M}^{\Pi_{s,\eta}}(1^k) = 1\right] - \Pr\left[\mathcal{M}^{\mathcal{U}_{k+1}}(1^k) = 1\right]\right| \geq \delta \ .$$

Then there is an algorithm $\mathcal{A}$ making $q' = O(q \cdot \delta^{-2}\log k)$ oracle queries, running in time $T' = O(T \cdot k\delta^{-2}\log k)$, and such that

$$\Pr\left[\mathbf{s} \leftarrow \{0,1\}^k \ : \ \mathcal{A}^{\Pi_{s,\eta}}(1^k) = \mathbf{s}\right] \geq \frac{\delta}{4} \ .$$


Encryption
Keystream Generator
secret key
Mapping
{x_i}
{y_i}
{a_i}
{b_i}
Error-Correction Encoding
plaintext
Wire-Tap Channel Encoding
{z_i}
{v_i}
{u_i}
PR-box
Source of Randomness
RAND-box
Public Comm. Channel
Decryption
PR-box
{a_i}
Error-Correction Decoding
plaintext
Wire-Tap Channel Decoding
{x_i}
{y_i}
Keystream Generator
secret key
Mapping

Algebraic Model of Stream Cipher II under Chosen Plaintext Attack

Corollary 1. Under the chosen plaintext attack which for each t implies $\mathbf{b}_t = \mathbf{0}$ (i.e. the all zeros vector), Proposition 1 implies:

$$\mathbf{z}_t = \mathbf{q}_t \mathbf{S} \oplus \vec{\nu_t} \ , \tag{1}$$

where

$$\mathbf{q}_t = (\bigoplus_{i=1}^{m} x_i \mathbf{h}_i \oplus \mathbf{y}_t)\mathbf{S}^{-1} \ \ , \ \ \vec{\nu_t} = (\bigoplus_{i=1}^{n-m} u_i \mathbf{g}_i) \oplus (\bigoplus_{i=1}^{m} v_i \mathbf{h}_i) \ , \tag{2}$$

and where $\mathbf{S}$ is an $n \times n$ binary matrix determined by the length k binary secret key $\mathbf{k}$, and $\mathbf{S}^{-1}$ is its inverse.

Assumption 1. For any $t = 1, 2, ...,$ $\vec{\nu_t} \leftarrow \mathrm{Ber}_{n,\eta}$, where η is the parameter.

A Statement on Stream Cipher II Security

Theorem 1. Assume there is an adversary $\mathcal{A}$, running in time T , and attacking the Stream cipher II specified by Corollary 1 and Assumption 1 with parameters (ℓ, m, k, n, η), $k = n$, in the sense of IND with advantage δ by making at most q queries to the encryption oracle. Then there is an algorithm $\mathcal{M}$ making $O(q)$ oracle queries, running in time $O(T)$, and such that

$$\left|\Pr\left[\mathrm{s} \leftarrow \{0,1\}^k \ : \ \mathcal{M}^{\Pi_{s,\eta}}(1^k) = 1\right] - \Pr\left[\mathcal{M}^{\mathcal{U}_{k+1}}(1^k) = 1\right]\right| \geq \frac{\delta}{n} \,. \tag{1}$$

VI. Concluding Notes

Main Messages of This Talk

- The talk has pointed out the **underlying algebraic structure** of certain stream ciphers which involve randomness and dedicated homophonic encoding.
- The algebraic model of the considered stream ciphers imply that **the security originates from the hardness of the LPN problem**.

Thank You Very Much for the Attention,

and

QUESTIONS Please!