Provably secure counter mode with related key-based internal re-keying

Goncharenko K.

Moscow State University, al_tair94@mail.ru,

Alekseev E.

CryptoPro LLC, Ph.D., alekseev@cryptopro.ru,

Marshalko G.

Technical committee for standardization (TC26), marshalko_gb@tc26.ru.

May 30, 2018

It is very important to analyse block ciphers in related-key model!

Why?

Because situation when a lot of key relations are known can emerge when key generating machine is broken. Or just key generation algorithm is poor (like it was in WEP).

One can consider this motivation to be inconvincing and the problem of related-key analysis to be just an entertaining task.

There are much less papers on this topic than on classic methods.

As a result we still have no published related-keys analysis on Kuznyechik!


НЕТ!

However the motivation can be adjusted with more real examples, when related keys appear (Razali E., Phan R.C.-W.: *On The Existence of Related-Key Oracles in Cryptosystems based on Block Ciphers*).

- the adversary can mount a "man in the middle" attack on the key transfer protocol that does not provide integrity;
- in some systems keys can be produced not equiprobable or be selected from a small set (it happened with WEP);
- if the hash function is an iterative construction built upon the blockcipher then the adversary also has a possibility of manipulating the encryption keys (e.g. Davies-Meyer scheme);

However the motivation can be adjusted with more real examples, when related keys appear (Razali E., Phan R.C.-W.: *On The Existence of Related-Key Oracles in Cryptosystems based on Block Ciphers*).

- the adversary can mount a "man in the middle" attack on the key transfer protocol that does not provide integrity;
- in some systems keys can be produced not equiprobable or be selected from a small set (it happened with WEP);
- if the hash function is an iterative construction built upon the blockcipher then the adversary also has a possibility of manipulating the encryption keys (e.g. Davies-Meyer scheme);
- some protocols may use related keys by design to generate key material easier and faster (such a cryptosystem is discussed in our work).

What about cryptographic synthesis motivation?

To answer this question let's look at related-key security from provable security perspective.

- We already got used to statements like "cryptosystem's security is based on hardness of discrete logarithm problem in the cyclic group".
- Block cipher could be considered as analogue of the cyclic group.
- In this case discrete logarithm problem could be turned into PRP-distinguishing problem.

- We already got used to statements like "cryptosystem's security is based on hardness of discrete logarithm problem in the cyclic group".
- Block cipher could be considered as analogue of the cyclic group.
- In this case discrete logarithm problem could be turned into PRP-distinguishing problem.
- Hardness of distinguishing in related-key adversary model is just another property of block cipher.

- We already got used to statements like "cryptosystem's security is based on hardness of discrete logarithm problem in the cyclic group".
- Block cipher could be considered as analogue of the cyclic group.
- In this case discrete logarithm problem could be turned into PRP-distinguishing problem.
- Hardness of distinguishing in related-key adversary model is just another property of block cipher.
- So we can consctruct cryptosystems which security is based on this property.

Related keys in algorithms

We state that related keys can be not only an attack method, but also a tool in arsenal for building new protocols.
This can be useful:

- Enlarge available tools for synthesis;
- Leads to better security estimations in various models;
- Proofs become shorter and clearer.

CTRR mode

We present CTRR ("CounTer with Related-key Re-keying encryption mode").

By this example we show:

- Possible construction of secure cryptosystem using related keys.
- Another way to solve problem of appearing of next section key in ciphertext.
- Way to estimate encryption mode security through security in related-key adversary model.

Let $2|n$, $n|k$ and $C_1, \ldots, C_{k/n} \in V_n$ are pairwise different constant bit strings. Parameters of the mode is section size N such that $n|N$ and transformation $\phi : V_k \to V_k$. The processing of the plaintext P by means of the initialization vector $IV \in V_{n/2}$ and key $K \in V_k$ is carried out as follows:

CTRR mode

1. $ctr_1 = IV||0^{n/2}$, $ctr_j = Inc(ctr_{j-1})$, $j = 2, \ldots, |P|_n$;
2. $K_1 = K$, $K_j = E_{\phi(K_{j-1})}(C_1)|| \ldots ||E_{\phi(K_{j-1})}(C_{k/n})$, $j = 2, \ldots, \lceil |P|/N \rceil$;
3. $G_j = E_{K_i}(ctr_j)$, $j = 1, \ldots, |P|_n$, $i = \lceil j \cdot n/N \rceil$;
4. Ciphertext C is equal to $P \oplus \mathrm{msb}_{|P|}(G_1||...||G_{|P|_n})$.

CTRR mode

Briefly stated, the plaintext is processed in the same way as in standard CTR mode but the key that is used to generate the encryption blocks G_j is not constant and is updated after generating N/n blocks G_j. The ϕ transformation can be any of the following:

- $\phi(X) = X \oplus c$, for some constant $c \in V_k, c \neq 0^k$;
- $\phi(X) = str_k(int(X) + c \ mod \ 2^k)$, for some constant $c \in \{1, \ldots, 2^k - 1\}$.

For the rest of the paper we will denote this set of possible variants for ϕ as $\hat{\Phi}$.

CTRR mode

The security analysis of the CTRR mode has been carried out in the IND-CPNA ("Indistinguishability under Chosen Plaintext and Nonce Attack") model. Informally, in this model the adversary has to distinguish the obtained ciphertexts from a random string, having the capability to adaptively choose plaintexts and nonces (in a unique manner).

For security estimation we need some more models.

CTRR mode

Standard "single-key" security notion for block cipher is PRP-CPA ("Pseudo Random Permutation under Chosen Plaintext Attack").

- Adversary $\mathcal{A}$ has access to the oracle $\mathcal{O}^{PRP}$ that takes blocks $M \in V_n$ as queries.
- $\mathcal{O}^{PRP}$ either implements permutation $P \in_{\mathcal{U}} Perm(V_n)$ or chooses key $K \in_{\mathcal{U}} V_k$.
- As a response to a query M oracle returns a string $P(M)$ or string $E_K(M)$ correspondingly.

Let bit b be equal 1 in the first case and 0 in the other. Adversary's advantage is calculated as

$$\mathrm{Adv}_{\mathsf{E}}^{\text{PRP-CPA}}(\mathcal{A}) = \mathbb{P}(\mathcal{A} \to 1 \mid b = 1) - \mathbb{P}(\mathcal{A} \to 1 \mid b = 0).$$

CTRR mode

An analogue of PRP-CPA notion for several related keys is PRP-RKA ("Pseudo Random Permutation under Related-Key Attack").
This model has an additional parameter, which is the set Φ of key-derivation functions $\phi : V_k \to V_k$.

- Adversary $\mathcal{A}$ has access to oracle $\mathcal{O}^{\text{RKA}}$ which takes pairs (ϕ, M), $\phi \in \Phi$ and $M \in V_n$ as queries.
- $\mathcal{O}^{\text{RKA}}$ chooses either family of permutations $G \in_{\mathcal{U}} Perm(V_k, V_n)$ and a key $K \in_{\mathcal{U}} V_k$ or only a key $K \in_{\mathcal{U}} V_k$.
- Oracle answers query (ϕ, M) with a string $G_{\phi(K)}(M)$ or a string $E_{\phi(K)}(M)$ correspondingly.

Let bit b be equal 1 in the first case and 0 in the other. Adversary's advantage is calculated as

$$\text{Adv}^{\text{PRP-RKA}}_{\Phi,\text{E}}(\mathcal{A}) = \mathbb{P}(\mathcal{A} \to 1 \mid b = 1) - \mathbb{P}(\mathcal{A} \to 1 \mid b = 0).$$

CTRR mode

Theorem (CTRR sequrity estimation, informal)

Let $N \in \mathbb{N}, \phi \in \Phi$ be parameters of CTRR mode.
Let adversary $\mathcal{A}$ attack CTRR in IND-CPNA task.
Let also m be the maximal message length and σ_i be number of blocks processed under i-th key.
Then there exists an adversary $\mathcal{B}$ solving PRP-RKA task such that

$$\mathrm{Adv}^{\mathit{IND\text{-}CPNA}}_{\mathit{CTRRN},\phi}(\mathcal{A}) \leqslant \ell \cdot \mathrm{Adv}^{\mathit{PRP\text{-}RKA}}_{\{\mathit{id},\phi\},E}(\mathcal{B}) + \\ + \frac{(\sigma_1+2)^2 + \ldots + (\sigma_{\ell-1}+2)^2 + (\sigma_\ell)^2}{2^n}$$

where $\ell = \lceil m/N \rceil$.

Problem of appearing of next section key in ciphertext is much more actual for modes with unpredictable block cipher input, such as CBC, CFB, OMAC modes (in OMAC-ACPKM-Master mode this problem is solved with internal re-keying with master key).

Therefore, using related keys for constructing internally re-keyed modes based on these ones probably may give much more useful impact.

This is the goal of future research.

Why the cipher is secure in PRP-CPA model? We believe so, based on great amount of research done.

The same amount of research should be done for related-key adversary model.

Kuznyechik in RK model

Let's find whether CTRR is compatible with Russian and international standards for block ciphers:

- AES: no. Its security was drastically decreased in [Biryukov, Khovratovich, 2009].
- Magma: no. Since in [Pudovkina, Khoruzhenko, 2013] there was presented an attack using related keys that recovered master-key in "sometimes acceptable time" $Adv^{\text{PRP-RKA}}$ for this cipher should be considered unacceptably big. But due to lax formulations Magma should be analysed in this model more.
- Kuznyechik: ???

Kuznyechik in RK model

No related-key cryptanalysis was published for Kuznyechik. We present first attempts of such analysis.

Kuznyechik in RK model

Reduced version of Kuznyechik block cipher has only 4 rounds of the basic cipher, and each round of the key schedule has only 2 rounds of basic cipher's Feistel rounds.
The key schedule of the reduced cipher uses round constants $\mathbf{C_1} = L(1), \mathbf{C_2} = L(2) \in V_{128}$.

Kuznyechik in RK model

Round keys $\mathbf{K_i} \in V_{128}$, $i = 1, 2, 3, 4$, are derived from

$$K = k_{255}||\ldots||k_0 \in V_{256},$$

$k_i \in V_1$, $i = 0, 1, \ldots, 255$, and evaluated according to the following equations:

$$K_1 = k_{255}||\ldots||K_{128};$$
$$K_2 = k_{127}||\ldots||k_0;$$
$$(K_3, K_4) = F[C_2]F[C_1](K_1, K_2).$$

Ciphertext obtained as a result of the following transformation.

$$E_{K_1,K_2}(m) = X[K_4]LSX[K_3]LSX[K_2]LSX[K_1](m)$$

Kuznyechik in RK model

Very brief description of attack:

- the attack consists of 2 parts: recovering left (K_1) and right (K_2) parts of master-key;
- K_1 is efficiently guessed and K_2 is easily obtained for known K_1;

- **key schedule and cipher itself both use the same LSX rounds**;

- **key schedule and cipher itself both use the same LSX rounds**;
- this can be used to force local collisions a round after injection point when the guess is right;

- **key schedule and cipher itself both use the same LSX rounds**;
- this can be used to force local collisions a round after injection point when the guess is right;
- due to ability inject differences in keys this injection point is the second round, not the first as it usually is in differential analysis;

- **key schedule and cipher itself both use the same LSX rounds**;
- this can be used to force local collisions a round after injection point when the guess is right;
- due to ability inject differences in keys this injection point is the second round, not the first as it usually is in differential analysis;
- **after 1 more round we can still easily detect local collision**;

- **key schedule and cipher itself both use the same LSX rounds**;
- this can be used to force local collisions a round after injection point when the guess is right;
- due to ability inject differences in keys this injection point is the second round, not the first as it usually is in differential analysis;
- **after 1 more round we can still easily detect local collision**;
- **we can inject differences in bytes independently,** therefore recovering K_1 not for 2^{128}, but for only $16 \cdot 2^8$.

- **key schedule and cipher itself both use the same LSX rounds**;
- this can be used to force local collisions a round after injection point when the guess is right;
- due to ability inject differences in keys this injection point is the second round, not the first as it usually is in differential analysis;
- **after 1 more round we can still easily detect local collision**;
- **we can inject differences in bytes independently,** therefore recovering K_1 not for 2^{128}, but for only $16 \cdot 2^8$.
- Got K_1!

- **key schedule and cipher itself both use the same LSX rounds**;

- **key schedule and cipher itself both use the same LSX rounds**;
- knowing K_1 and controlling plaintext we can force known inner state before the last round that doesn't include K_2;

- **key schedule and cipher itself both use the same LSX rounds**;
- knowing K_1 and controlling plaintext we can force known inner state before the last round that doesn't include K_2;
- so after the last round (in ciphertext) K_2 will be a linear part of the sum.

- **key schedule and cipher itself both use the same LSX rounds**;
- knowing K_1 and controlling plaintext we can force known inner state before the last round that doesn't include K_2;
- so after the last round (in ciphertext) K_2 will be a linear part of the sum.
- Got K_2!

Kuznyechik in RK model

Overall time complexity of the attack is 2^{12} encryptions and 2^{12} related keys.
Our non-optimized implementation of the proposed attack implementation in Python 2.7 on a standard PC required approximately 10 minutes to recover the correct key.

Kuznyechik in RK model

- The feasibility of the proposed attack is based mainly on the extremely simplified key schedule of a reduced version of the Kuznyechik, which allows the attacker to trace the desired differences in round keys.

Kuznyechik in RK model

- The feasibility of the proposed attack is based mainly on the extremely simplified key schedule of a reduced version of the Kuznyechik, which allows the attacker to trace the desired differences in round keys.
- The main problem of using related-key approach is high diffusion properties of Kuznyechik round function.

Kuznyechik in RK model

- The feasibility of the proposed attack is based mainly on the extremely simplified key schedule of a reduced version of the Kuznyechik, which allows the attacker to trace the desired differences in round keys.
- The main problem of using related-key approach is high diffusion properties of Kuznyechik round function.
- Enhancing attack for even 1 additional round in key schedule or cipher itself needs using complex statistical properties of round functions. Enhancing attack to full cipher doesn't seem to be possible.

Demonstrated attack rather shows good security in related-key model of full Kuznyechik than bad.

Based on it we can make first assumptions of negligibility of it's $InSec^{\text{PRP-RKA}}$. **But a lot more research should be done to gain some level of confidence in it.**

Conclusion

- Related keys can be used not only as attack vector, but as a feature of cryptographic algorithm;
- Related-key cryptanalysis of modern ciphers should become more popular

Thanks for your attention!