Contents

1 PKE Algorithms

1.1 Some types of PKE

There is a list of different crypto-algorithms based on Diffie-Hellman protocol:

1. PKEET;
2. PCE;
3. AoN-PKEET;
4. FG-PKEET;
5. PKE-DET;
6. PKE-AET;

Table 1 shows the comparison of these encryption methods.

Table 1: The comparison of encryption methods

		PKEET	PCE	FG-PKEET	PKE-DET	PKE-AET
Efficiency	KeyGen	TE	TE	2TE	2TE	2TE
	Enc	3TE	4TE	4TE	TP+5TE	4TE
	Dec	3TE	2TP+TE	2TE	TP+4TE	4TE
	Aut	–	–	3TE	3TE	2TE/3TE
	Test	2TP	4TP	4TP	4TP+2TE	2TP+(4TE/2TE)

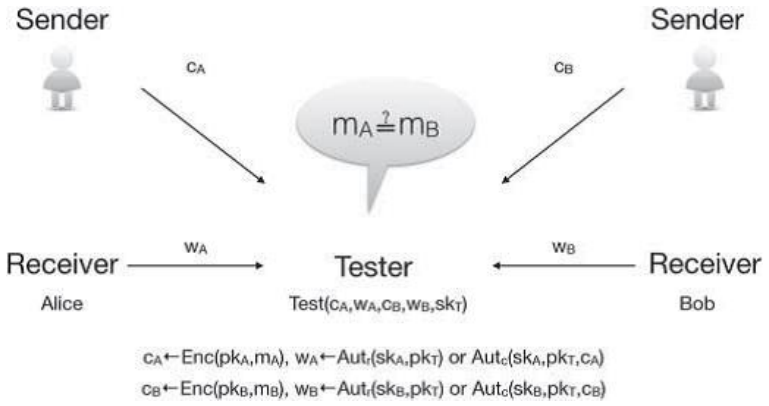

Figure 1: PKEET algorithm illustration

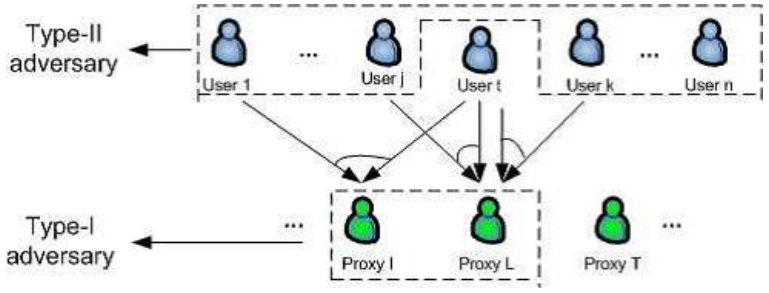

Figure 2: FG-PKEET algorithm illustration

Figure 1 and Figure 2 shows, how works PKEET algorithms.

1.2 Advantages of PKEET (Public Key Encryption with Equality Test) encryption methods:

- Security: PKEET provides robust security through its asymmetric encryption approach, ensuring that data remains confidential even if the public key is known.

- Ease of Key Management: Unlike traditional symmetric encryption methods, PKEET eliminates the need for secure key exchange between parties, as each entity has its own pair of public and private keys.

- Authentication: PKEET enables authentication of both parties involved in communication, as each participant can verify the identity of the other through their digital signatures.

- Non-Repudiation: PKEET facilitates non-repudiation, meaning that neither party can deny their involvement in the communication or transaction, as their digital signatures provide undeniable proof of participation.

- Flexibility: PKEET allows for flexible usage in various cryptographic protocols and applications, including secure communication, digital signatures, and key exchange mechanisms.

- Resistance to Quantum Attacks: Some PKEET implementations, such as those based on lattice-based cryptography, offer resistance to quantum computing attacks, ensuring long-term security even in the face of evolving threats.

1.3 Drawbacks of PKEET (Public Key Encryption with Equality Test) encryption methods:

$$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \left[\frac{-\hbar^2}{2m}\nabla^2 + V(\mathbf{r},t)\right]\Psi(\mathbf{r},t) + \int d^3\mathbf{r}' \int dt'\, \Sigma(\mathbf{r},\mathbf{r}',t,t')\Psi(\mathbf{r}',t')$$

- Computational Overhead: PKEET encryption typically involves complex mathematical operations, leading to increased computational overhead compared to symmetric encryption methods. This can impact performance, especially in resource-constrained environments.

- Key Management Complexity: Managing a large number of public and private key pairs in PKEET systems can be challenging, especially in scenarios with multiple users or devices. Key generation, distribution, and revocation require careful coordination and infrastructure.

- Vulnerability to Side-Channel Attacks: PKEET implementations may be susceptible to side-channel attacks, where an attacker exploits physical properties of the cryptographic system (such as power consumption or timing) to extract sensitive information.

- Limited Efficiency for Large Data: While PKEET encryption is effective for securing small pieces of data, such as cryptographic keys or digital signatures, it may not be as efficient for encrypting large volumes of data due to the overhead associated with asymmetric encryption algorithms.

- Dependency on Trust Infrastructure: PKEET relies on a trusted infrastructure for key management and authentication, including the availability and integrity of public key repositories and certification authorities. Compromises in this infrastructure can undermine the security of the entire system.

- Potential for Key Compromise: Like any cryptographic system, PKEET is vulnerable to key compromise if an attacker gains unauthorized access to a user's private key. This can lead to unauthorized decryption of sensitive information or impersonation attacks.

2 Pollard's lambda-method

$$\begin{aligned}
\nabla \cdot \mathbf{E} &= \frac{\rho}{\varepsilon_0} \\
\nabla \cdot \mathbf{B} &= 0 \\
\nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t} \\
\nabla \times \mathbf{B} &= \mu_0 \left(\mathbf{J} + \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t} \right)
\end{aligned}$$

Pollard's lambda-method allows to find a discrete logarithm that is known to lie in a fixed interval [A; B] of length $w = B - A$ in heuristic expected time $O(\sqrt{w})$ (instead of $O(\sqrt{w} * \log w)$ with a simple generalization of the baby-step giant-step method). The idea is to compute two sequences of group elements, one starting with the upper limit B of the interval (the "trail of the tame kangaroo") and the other with the group element y of which the discrete logarithm should be computed (the "trail of the wild kangaroo"). The behavior of both sequences is given by

$x_{i+1} = x_i * h^{f(x_i)}$ where f is a "random-like" function taking integer values in a range R of mean m, where $m = a * w^{(1/2)}$ for some a depending on the tolerated failure probability

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi)\, e^{2\pi i \xi x}\, d\xi = \sum_{n=-\infty}^{\infty} f_n\, e^{2\pi i n x/T},$$

where $\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i \xi x}\, dx$ is the Fourier transform of $f(x)$, and $f_n = \frac{1}{T}\int_{-T/2}^{T/2} f(x) e^{-2\pi i n x/T}\, dx$ are the Fourier series coefficients.

The starting point of the trail of the tame kangaroo (the sequence $x_0, x_1, ...$) is $x_0 = h^B$, and the group elements $x_0, x_1, ..., x_n$ are computed (for some fixed N). The trail of the wild kangaroo (the sequence $x_0, x_1, ...$) starts at $x'_0 = y$ and stops with x'_M if condition 1 is met

$$\sum_{j=0}^{M-1} f(x') > \sum_{i=0}^{N-1} f(x_i) + (B - A) \tag{1}$$